Q # 1046

ORNL/NUREG/TM-99

ORCODE.77

A Computer Routine to Control a Nuclear Physics Experiment by a PDP-15 + CAMAC System, Written in Assembler Language and Including Many New Routines of General Interest

J. K. Dickens

J. W. McConnell

MASTER

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Under Interagency Agreements ERDA 40-551-75 and 40-552-75

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy \$5.50; Microfiche \$3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration/United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Contract No. W-7405-eng-26

Neutron Physics Division

ORCODE.77

**A COMPUTER ROUTINE TO CONTROL A NUCLEAR PHYSICS EXPERIMENT
BY A PDP-15 + CAMAC SYSTEM, WRITTEN IN ASSEMBLER LANGUAGE
AND INCLUDING MANY NEW ROUTINES OF GENERAL INTEREST**

J. K. Dickens and J. W. McConnell

Manuscript Completed - March 29, 1977

Date Published - May 1977

Prepared for the
U. S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Under Interagency Agreements ERDA 40-551-75 and 40-552-75

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

CONTENTS

	<u>Page</u>
Abstract	1
Introduction	2
General Information and Operation	4
Description of ORCODE.77 Routines	9
A. Beginning of Listing: Symbols and Mnemonics	13
B. Starting Execution	15
C. Display Routine	17
D. Executive Routine to Handle Teletype Commands	23
E. Interrupt Routine	26
F. CAMAC Routines	27
G. Teletype Input and Output Routines	31
H. Type Out Contents of Channel in AC	37
I. Scaler Routines	38
J. Start and End of Run Routines	40
K. DECTAPE Routines	42
L. Data Manipulation; Integration and Partial Type Out	52
M. DECTAPE Directory	54
N. Initializing and Bookkeeping Routines	55
O. Filename and Extension Routines	58
P. Clock Interrupt	60
Q. Processor and Data Overflow List	62
R. Read Data from or Write Data onto DECTAPE	64
S. Direct Dump to PDP-10	65

CONTENTS

	<u>Page</u>
T. Floating-Point Routines	71
U. Nuclear Data 5050 Analyzer Dump	74
V. Definition and Storage of Constants	81
Acknowledgments	85
Appendix. CAMAC Instructions and Programming Examples	86
References	93

ABSTRACT

ORCODE.77 is a versatile data-handling computer routine written in MACRO (assembler) language for a PDP-15 computer with EAE (extended arithmetic capability) connected to a CAMAC interface. The Interrupt feature of the computer is utilized. Although the code is oriented for a specific experimental problem, there are many routines of general interest, including a CAMAC Scaler handler, an executive routine to interpret and act upon three-character teletype commands, concise routines to type out double-precision integers (both octal and decimal) and floating-point numbers and to read in integers and floating-point numbers, a routine to convert to and from PDP-15 FORTRAN-IV floating-point format, a routine to handle clock interrupts, and our own DECTAPE handling routine. Routines having specific applications which are applicable to other very similar applications include a display routine using CAMAC instructions, control of external mechanical equipment using CAMAC instructions, storage of data from an Analog-to-digital Converter, analysis of stored data into time-dependent pulse-height spectra, and a routine to read the contents of a Nuclear Data 5050 Analyzer and to prepare DECTAPE output of these data for subsequent analysis by a code written in PDP-15-compiled FORTRAN-IV.

INTRODUCTION

ORCODE.77 is an experiment-controlling computer routine, written in MACRO-15 Assembler Language,¹ designed to maximize control over an activation type experiment utilizing a PDP-15 computer for data input, manipulation, storage, and output. The experiment consists of placing a sample in a carrier (known as a rabbit), transporting the sample via a pneumatic tube to an irradiation facility, returning the sample to a counting position, and counting the emanations from the sample. A schematic of the hardware used is illustrated in Fig. 1 for the present experiment. Not shown directly in this figure is a link between the CAMAC interface and the pneumatic tube controller.

The central processor (CP) is a PDP-15 computer having 24K 18-bit words of memory. In addition to standard peripheral equipment shown, there is a link through the CAMAC interface to a telephone wire for direct transmission of data to another PDP-15, which in turn transmits the data to a PDP-10 for storage on disk and later processing.

The experiment involves preparing a sample in a rabbit and placing it in the sample holder in Hood No. 1. Then a series of instructions, including the desired irradiation time and up to 17 counting-time intervals, are entered via teletype. To begin an experimental run, the computer sends a signal to initiate the sample movement to the irradiation position; the irradiation period is monitored by the computer, and a signal is sent to initiate the return of the sample after the irradiation time has elapsed. The rabbit returns to Hood No. 2, and cools for a designated period. The cooling time must be ≥ 2 sec, but can be as long as desired. Upon initiation of the counting time, data from the analog-to-digital

Fig. 1. Schematic representation of experimental arrangement for fission-product decay-heat measurements. Samples of ^{235}U (or ^{239}Pu) are placed in a rabbit and put in the Sample Sender. The Central Processor controls the movement of the rabbit to and from the Irradiation Position, then to the Sample Holder. After a specified cooling time, either photons or beta rays are counted, and the data are stored in the Central Processor. At the completion of data accumulation the data are stored on magnetic tape for off-line reduction.

converter (ADC) in coincidence with, or lacking, an identification (Tag) bit, are read into a preset storage area in the computer. During the course of the experiment many such data are transferred to the computer. The CP processes the data into 512-channel spectra which are stored in the computer memory. During data accumulation, data may be displayed. At the end of the data accumulation, data are stored on DECTAPE for transfer to a PDP-10 for processing. During off-hours the data may be transferred to the PDP-10 via telephone wires.

Other than the computer and standard peripherals, the only hardware of concern is the CAMAC crate control unit. This unit was designed and fabricated at ORNL according to standard specifications listed in the several TID reports on CAMAC instrumentation.²⁻⁴

GENERAL INFORMATION AND OPERATION

The basic purpose of the experiment is to obtain spectra of beta or gamma rays emitted by a sample as a function of the time following thermal-neutron irradiation (activation). Once the sample has been placed in the rabbit and the rabbit placed in position in Hood No. 1, the CP controls the subsequent rabbit movements and data acquisition. To do this the desired parameters are entered into computer memory through the use of the command `3KK_` [note: to have a command take effect, the three characters must be followed by a space () or a carriage return (+)]. Figure 2 illustrates an example of parameter input. The underlined portions are the typed-in entries by the user; the portions not underlined are typed out by the CP. In this routine a comma (,) is used as a number delimiter.

```

BKK+
INPUT DATA LOG
235-U B-21 1UGM G=8 22:25 9-JULY+
RUN NUMBER = 2821,JKD
IRRADIATION TIME = 100.7,
NO OF TIME INTERVALS = 15,
70, 110, 170, 250, 350, 550, 750, 1150, 1550, 1950,
2450, 2950, 3950, 5950, 9950, 13950,
S(OURCE) OR R(ABBIT)? 0

```

Fig. 2. Example of initiation of data taking; user entries are underlined. BKK+ initiates the bookkeeping routine. The next user entry is an identification title. The next entry is a run number, followed by the irradiation time in sec. In this example the sample cooling time is 70 sec, and the first of 15 counting intervals is 40 sec.

We follow this example in detail. After receiving the command BKK+ the CP asks for a title. We type in the title (≤ 40 characters) describing the experiment. Then the CP asks for an identification number (≤ 6 digits) and assigns the current extension to this ID number for a DECTAPE filename. Then the irradiation time is asked for and entered (in sec). Then the number n of (consecutive) counting time intervals are entered, and then $n + 1$ times (in sec) are entered. The first of these entries is the cooling time. These numbers must increase or an error message is typed out, and the task is redone. Lastly, either R or S is required; if R, the rabbit transport system is initiated; if S, there is no pneumatic operation.

The experiment is initiated by entering the command STR_. After transfer of the rabbit to and from the irradiation position, the first counting interval is between 70 and 110 sec after the end of irradiation; the second counting interval starts at the end of the first; the third counting interval starts at the end of the second; etc. At the end of the last counting interval the CP reports END OF RUN.

Each count registered in the detector is processed by the electronics and the output of the ADC, a pattern of 9 bits, and the Tag pulse, if present, are transmitted to the computer by the CAMAC interface. The data transmission takes place without interrupting the program counter, on a cycle-stealing basis. When a 256-word data buffer is full, a program interrupt occurs causing a new buffer region to be defined. The CP then sorts the 256 data words into two 512-channel spectra corresponding to the tag bit and adds these spectra to previously analyzed data. At the end of the first counting period a program interrupt occurs causing a new input

buffer region to be defined. The CP processes the partially-filled old buffer adding to results already in core. Then the first-word-address of the spectrum storage region is incremented by 1024 locations.

In this manner spectra are obtained for up to 17 consecutive time intervals; most of the 24K-word storage is used to store accumulated data.

When the CP is not otherwise engaged, a portion of the data-accumulation region is displayed. The display and other electronics is shown in Fig. 3. At the top is the display controller unit and just below it is the display. The six push buttons on the left of the controller are referred to in the code as Buttons 1 - 6. The four push buttons in the center control movements of displayed vertical lines. The left-most multiple switch controls the full scale of the display and the right-most switch controls the choice of region(s) to be displayed. The CAMAC interface is used to transmit trace coordinates to an oscilloscope, and to stobe the status of ten push buttons and two multiple position switches and transmit this status to the CP upon command. This status determines which of the 34 512-channel regions to display, the horizontal and vertical scales, and display and movement of vertically displayed lines which are used to define regions of interest. The CP operates in the display loop essentially all of the time; count rates > 20000/sec will affect the display continuity sufficiently to be observable. All, or a portion, of a second data-accumulation region may be simultaneously displayed (overlap display).

Primary data storage is on DECTAPE; however, at certain times the computer may act as a remote terminal to a PDP-10 (through other peripheral

Fig. 3. Shown in this photograph are, from top to bottom, (a) the display control box, (b) the display oscilloscope, (c) two NIM bins containing modular electronics, and (d) the CAMAC bin containing: 12 scalars at the left of the bin, in the center are the control units for the ADC, the display, and the telephone transmission, and to the right is the interface module to the PDP-15.

computers) and transmit data to specified disk storage. This option requires a filename and a project-programmer number which are entered via the teletype.

Data on DECTAPE may be read into the computer. All or some of the data dumped with a given filename may be retrieved. A partial readin may be routed into a desired part of the data storage area to facilitate overlap display with data from a different run.

The various subprograms are initiated by entering a 3-character command via the teletype. These are summarized in Table 1; the pertinent part of the code is listed on pages 24 and 25. Some operations render other commands inoperable until completion of the desired operation. For example, writing on a DECTAPE prohibits any other DECTAPE command. The important subroutines for this function are titled "LOCK" and "UNLOCK" and are listed on page 25.

DESCRIPTION OF ORCODE.77 ROUTINES

In this section we present a complete listing of the code as of 1 Feb. 77, and discussions of the main sections. We dispense with the traditional flowcharts, because neither author uses flowcharts in preparing a computer routine. Under the circumstances the lack of flowcharts has the advantage of not presenting an incorrect chart determined a posteriori from correct coding, but has the disadvantage of requiring more careful attention if part of this code is to be adapted for other purposes.

The code is listed in the order which it is currently written, which is not necessarily optimal. The listing is broken up into sections, each

with a discussion of the section. Routines of general interest include Teletype Commands (Sect. D), Teletype Input and Output (Sect. G), Data Manipulation (Sect. L), and the new Floating Point Routines (Sect. T). The Interrupt Routine (Sect. E) should be of interest for users of this feature. Users of CAMAC I/O should first read the Appendix.

TABLE 1. TYPE-IN INSTRUCTIONS

Command	Subprogram performed.
BKK	Receive from teletype and store information and parameters for the next data-taking period.
BUN	Clear status of rabbit transport system.
DDT	Jump to DDT for debugging. Not used in full version of code.
DEC	Set for decimal input and output.
DIR	Type out DECTAPE directory.
DMP	Dump ID and spectral data on DECTAPE.
ETA	(Last 3 letters of BETA.) Set rabbit transport system for beta-ray experiment.
GET	Get information from DECTAPE.
HIL	Set vertical display line at channel typed in.
HI=	Type channel no. and contents of the channel corresponding to the right-most vertical display line.
HLT	Halt data accumulation.
INT	Integrate data displayed between vertical lines. (Centroid is obtained also.)
JFE	Dump contents of Nuclear Data 5050 Analyzer onto DECTAPE along with typed-in information.
KLN	Kill vertical lines on the display.
LNS	Type channel numbers and contents for both displayed lines.
LOL	Set a second vertical display line at typed-in channel.
LO=	Type channel number and contents of the channel corresponding to the left vertical line.
MMA	(Last 3 letters of GAMMA.) Set rabbit transport system for gamma-ray experiment.
OCT	Set for octal input.

TABLE 1. TYPE-IN INSTRUCTIONS (continued)

OVR	Type out overflow list, a table of channel numbers, and number of overflows.
RUN	Type out current run number.
SCA	Type out scaler information.
SOU	Start a one-spectrum run without BKK. Used for sources during set-up procedures.
STR	Start data run.
TEN	Send data to PDP-10.
TOL	Type out data displayed between vertical lines.
TS=	Type out current time (since end of irradiation).
TYI	Enter N integers into memory starting at absolute location M. Not often used.
XIN	Change the current filename extension.

A. Beginning of Listing: Symbols and Mnemonics

Symbols relating to instructions for the Nuclear Data 5050 Analyzer are listed first. Not all of them are used. Note particularly the mnemonics for 5 often-used commands; (a) LAM= -1, (b) EXIT= leave sub-routine, (c) LDI= load indirect, (d) STI= store indirect, and (e) ZAP= clear accumulator and link.

```

/ JOLLY GREEN GIANT SYMBOLS
/ BUNNY COMMANDS NOW IN (FEBRUARY 27TH)
/
/ TEST AND CONTROL
/
DMSF=783783 /SKIP IF NOT BUSY AND TURN COMPUTER ON-LINE
CONN=783722 /TURN COMPUTER ON-LINE
COFF=783724 /TURN COMPUTER OFF-LINE
AOFF=783822 /TURN ANALYZER OFF
AONN=783824 /TURN ANALYZER ON
/ ADDRESS REGISTER
/
CSAR=783463 /LOAD ADDRESS REGISTER
CSARR=783467 /LOAD ADDRESS REGISTER AND READ/RESTORE
/ 18 BIT DATA TRANSFERS BETWEEN ANALYZER AND COMPUTER
/
RQMR=783512 /READ MEMORY REGISTER
RRMR=783524 /+1 TO ADDRESS REGISTER AND READ/RESTORE
/
/ 12 BIT TRANSFERS
/ MOST - BETWEEN AC6-AC17 AND MEMORY REGISTER MR11-MR18
/ LEAST - BETWEEN AC6-AC17 AND MEMORY REGISTER MR23-MR32
/
RQMH=783532 /READ MEMORY MOST
RQML=783552 /READ MEMORY LEAST
/
/ DECTAPE
DTXA=787544;DTLA=787545;OTRA=787552;DIEF=787561;OTRB=787572;OTDF=787681
/ CENTRONICS LINE PRINTER
LBSF=786571 /SKIP ON LP DONE
LPCD=786621 /CLEAR LP DONE
/ DEC
/ CAMAC CONSTANTS
SCAN=1*16 /FIRST SCALER
ASCHN=18*16 /COMMUNICATIONS MODULE
DSPN=13*16 /DISPLAY MODULE
ADCN=14*16 /ADC MODULE
BUNN=17*16 /BUNNY CONTROLLER
DCHN=23*16 /DATA CHANNEL CONTROL
/
/ CAMAC FUNCTION CODES
F8=8
F101=512
F202=512
F808=512
F16=16*512
F17=17*512
F18=18*512
F19=19*512
F21=21*512
F23=23*512
F25=25*512
F27=27*512

```

```

/      CAMAC SUB-ADDRESSES
A0=0
A1=1
A2=2
A3=3
A4=4

```

```

A10=12
A13=13
A14=14

```

```

      .OCT
B10=100000 /DATA MODE
ENC=400000 /EXECUTE BIT

CAN=700000 /CRATE CONTROLLER IOT
/
LAN=777777
EXIT=JMP
LDI=LAC
STI=DAC
SAP=CLAI/CLL

MEMMAX=17777
NUMSPC=21
BUFSZ=400
DRAK=777 /FOR 512 CHANNELS
SPECSE=1000
BLOC1=14000 /FOR TEST CASE...SET =10000 WHEN READY
AGCN=40000
OVERMX=40 /MAXIMUM NO OF OVERFLOWS
DATBLK=OVERMX*2+32
DISBUF=50000 /DISPLAY BUFFER
/

```

B. Starting Execution

Normal starting location is 100; 40 is used during debugging to allow the debugging code DDT to occupy loc 16000 - 17777. Starting at either location resets all counters and locations, and clears all interrupts.

```

      .ASS
/
/STARTING HERE LEAVES SPACE FOR DDT
      .LOC 40
      LAC C20K
      DAC .SLOC1
      AAC-1
      DAC .SLOC1
      AAC-1
      DAC .SLOC2
      AAC 2
      TCA
      DAC .MSLOC
STARR LAC CHLT
      DAC 21
      JMP 100
CHLT  HLT
C20K  22000
      .LOC 100
/BOOTSTRAP ROUTINE TO SET UP INTERRUPT HANDLING
/AND TO INSURE THAT FIRST KEYBOARD INTERRUPT GOES TO THE EXECUTIVE ROUTINE
BEGIN CAP
      IOF
      EDA
      LAN-20
      DAC CODEND
      LAC .JMPIN
      DAC 1
      LAC .LACEX
      DAC INTYPE
      LAC .INDIT      /REMOVE DEC TAPE ERROR FLAG

```

/ ALSO READER FLAG BIT
/SET UP TABLE OF LOCATIONS

NUMH
DAC INTRK
-NUMSPC
DAC TALLY
LAC .TAB1
DAC 18
LAC .SCLR
SII 12
AAC 38
ISE TALLY
JMP .-3
-NUMSPC
DAC TALLY
LAC .SLUC1
DAC REG1
DAC REG2
SII 17
TAD SPTM
ISE TALLY
JMP .-3
D2M DTDFLG
D2M DTDFLG
D2M DTDFLG
D2M REG3
D2M REG4
D2M SF1
JMS PSIDE
LAC .JCLK
DAC ICLK
JMS 36IN
-7
JMS SETUP
LAC .MCC
CAN 24
LAC .ICAM
CAN 24
JMS
LAC
JMS -CH
ION

/FOR DECTAPE HANDLING

C. Display Routine

As mentioned earlier, the code spends most of its time in the display routine, leaving it only to process an interrupt. The display is standard at 512 channels, x-axis expansion is controlled by button 4 and vertical-line positions. Which of the 34 data storage areas to display is determined by the positions of the multiple selection switch (1 to 12), button 1 (off select 1 to 12, on select 13 - 17), and button 2 (tagged or untagged). After setting these buttons, pushing button 5 initiates transfer of data from the selected region to the primary display buffer, modified by scale factor determined by the other multiple selection switch (64 to 131072 full scale). Data from the primary display buffer is shifted to the secondary display buffer and may be displayed in overlap if button 4 is set. The data from the selected regions are continually scaled and transferred to the display buffer (i.e. live display).

```

/NEXT PORTION OF CODE IS MAIN ROUTINE
ARESET LAC ,CMB          /READ CONTROLLED BOX
      CAM 5
      JMP ,=1
      CAM 72
      DAC CBOX
      AND ,C1/
      DAC YFLSC
      LAC CBOX
      LRS+4
      AND ,C1/
      DAC RSELEC
/BUTTONS 1-4 ARE PUSH-ON, PUSH-OFF TYPE
/BUTTON 1 =0 FOR SELECT REGION 1-12
/          1 FOR SELECT REGION 13-MAX (18 PROBABLY)
/BUTTON 2 =2 FOR UNTAGGED SPECTRUM OF REGION
/          1 FOR TAGGED SPECTRUM
/BUTTON 3 =3 FOR SINGLE SPECTRUM DISPLAY
/          1 FOR OVERLAP
/BUTTON 4 =0 EXPAND BETWEEN LINES

```

/ 1 FULL SIZE DISPLAY
 / BUTTONS 5 AND 6 ARE MOMENTARY
 / BUTTON 5 TO LOAD INTO PRIMARY DISPLAY REGION SPECTRUM DETERMINED
 / BY THE SELECT SWITCH AND BUTTONS 1 AND 2

DZM BUT4
 LAC CBOX
 ALS+4
 DAC BUT6
 RAL
 DAC BUT5
 RAL
 SPA:RAL
 ISZ BUT4
 DAC BUT3
 RAL
 DAC BUT2
 RAL
 DAC BUT1
 LAC BUT4
 SMO BF1
 JMP .+7
 DAC BF1
 IOF
 SZA
 JMP EXPNO
 JMS FSIZE
 ION
 -4
 TAD YFLSC
 SMA
 JMP .+4
 TCA
 TAD .LLS
 SKP
 TAD .LRS
 DAC SHFWRD
 LAC BUT5
 SMA
 JMP DSPD
 LAC REG1
 DAC REG2
 LAC REG3
 DAC REG4
 LAC BUT1
 SPA:CLA
 AAC 14
 TAD PSELEC
 DAC REG3
 TAD NUMH
 SPA
 JMP .+3
 LAC ,NUM1
 DAC REG3
 LAC REG4
 TAD ,TABNU
 DAC REG1
 LOI REG1
 DAC REG1
 LAC BUT2
 SMA

/HAS SPECTRUM RELATIVE NUMBER FROM 0 TO NUMSPC-1

```

JMP +6
LAC .SPECS
TAD REG1
DAC REG1      /HAS FIRST LOCATION OF DATA TO MOVE TO DISPLAY
DSPG  LAC .DISBU
DAC 9F4
LAM
DAC 9F5
LAC REG1
DSPG1 DAC 9F6
LAC .SPECS
TCA
DAC 9TALLY
CLO
LDI 9F6
CLL
SHFWRD LLS+42
LMO
CLL
TAD .M200      /TEST FOR OVERFLOW
SNL
JMP +6          /LET DISPLAY OVERFLOW IF ON 131K SCALE ONLY
LAC YFLSC
SAD .C13
JMP +3
LAC .C177      /IF NOT ON 131K SCALE DISPLAY AT TOP OF SCPE
SKP
LACO
STI 8F4
ISZ 8F4
ISZ 8F6
ISZ 9TALLY
JMP SHFWRD+3
ISZ 8F5
JMP +3
LAC REG2
JMP DSPG1
DSPG2 LAC LINEHI
JMS DSPLNN
DAC LMA
LAC LINELO
JMS DSPLNN
DAC LLOA
LAC BUTJ
SHA:CLA
JME +3
LAC .SPECS
TAD SPECX
DAC SECSP
JMP ARESET
DSPLNN XX      /DISPLAY A VERTICAL LINE FOR GIVEN INPUT X
DAC XLN
LAC SPECX
AND .OMASK
TCA
TAD XLN
SPA
JMP DSPLNX
DAC XLN
TAD DISPLN      /DISPLN IS A NEGATIVE NUMBER

```



```

SNA
JMP DSPLNX
LAC XLN
SNA:CLL
EXIT DSPLNN
M'L
1700
CLL
DIV
SPEC2 1700
LACO
EXIT DSPLNN
DSPLNX CLC
EXIT DSPLNN
XLINES XX
JMS DSP
LAC .CLNSB
JMS CANAC2
JMS DSPOUT
JMP XLINES
DSPX DZM XINCR
LAC LHIA
SMA
JMS XLINES
LAC LLOA
SMA
JMS XLINES
LAC SPECX
DAC DISPLC
LAC INCR
DAC XINCR
CLA
JMS DSP
LAC .TAB11
JMS CANAC2
JMS DSPOUT
LAC SECSP
SNA
JMP DSPX
DAC DISPLC
CLA
JMS DSP
LAC .TAB11
JMS CANAC2
JMS DSPOUT
JMP DSPX
INCR 6
SECSP 8
/KILLS LINES
LINKIL CLC
DAC LINELO /SET BOTH = -1 TO KILL DISPLAY
DAC LINEMI
JMP DISHIS
ALO LAC LLOLINE /SETS LOW LINE FROM TELETYPE
JMS STRTYP /JMS STRASC,JMS TYPEIN,JMS CRLF
DAC LINELO
JMP DISHIS
AH1 LAC HILINE /SETS HIGH LINE FROM TELETYPE
JMS STRTYP
DAC LINEMI

```

```

EXPND    JMP DISNIS
          JMS LNSCHK      /DISPLAY PORTION BETWEEN LINES
          AAC 2
          DAC SPEC2
          JMS DISNO
          JMS DSPLOL
          DAC SPECX
          JMP VPY-1
DSPLOL   XX
          LAC SPECX
          AND .B77
          TAD LINELO
          EXIT DSPLOL
LNSCHK   XX      /CHECK ORDER OF LINES
          LAC LINEHI
          SPA
          D2M LINEHI
          LAC LINELO
          SPA
          D2M LINELO
          LAC LINELO
          CMA
          TAD LINEHI
          SMA
          EXIT LNSCHK    /EXIT WITH DIFFERENCE
          IAC
          SNL
          EXIT LNSCHK    /THIS JUMP IF LINELO=LINEHI
                          /IF LINELO .GT. LINEHI THEN REVERSE
          LAC LINELO
          DAC LSV
          LAC LINEHI
          DAC LINELO
          LAC LSV
          DAC LINEHI
          JMP LNSCHK+1
LNIA     2
LLOA     2
.CLNSB   LNSB+1
LNSB     11
          LINSB
          12
          -36
LINSB    4
.C100    100
          140
          200
          240
          300
          340
          400
          440
          500
          540
          600
          640
          700
          740
.C777    777
          1240
.C1100   1100
          1140
          1200
          1240
          1300
          1340
.C1400   1400
          1440
          1500
          1540
          1600
          1640
          1700
          1740

```

```

DSPOUT DSPX
      JMP CAMDIS
FSIZE  XX
      LAC ,DISBU
      DAC SPECX
      LAC ,SPECS
      DAC SPECE
      JMS DISNO
      LAC SECSP
      SNA
      EXIT FSIZE
      LAC SPECX
      TAD ,SPECS
      DAC SECSP
      EXIT FSIZE
MARKER LAC ,CHB
      CAM 5
      JMP ,+1
      LAC ,AAC1
      DAC LNPAT+2
      CAM 72
      RTL
      LMO
      LAC ,AACH1
      SZL
      DAC LNPAT+2
      LACQ
      SNA
      JMP ,+7
      JMS LNPAT
LINEH1 SPECS2-1
      LACQ
      RAL
      SNA
      JMP CAMDIS
      JMS LNPAT
LINELO 0
      JMP CAMDIS
      .AAC1 AAC 1
      .AACH1 AAC -1
      LNPAT XX
      LAC+ LNPAT
      XX
      SPAICLL
      CLA
      DAC+ LNPAT
      TAD MSPECE
      SPA
      JMP ,+3
      LAC ,C777
      DAC+ LNPAT
      ISZ LNPAT
      JMP+ LNPAT
MSPECE -SPECS2
DISNO  XX
      DAC DDSNO
      TCA
      DAC DISPLN
      LAC ,BTHOU
      CLL
      IDIV
      DDSNO 1200
      LACQ
      DAC INCR
      AAC-200
      SPA
      EXIT DISNO
      LAC ,C177
      DAC INCR
      EXIT DISNO

```

/DISPLAY FULL SIZE SPECTRUM
/REQUIRES RESETTING INFO TO CAMAC

/SHIFT MARKERS (LOW AND HI LINES)

/CLEARED MARKER INTERRUPT

D. Executive Routine to Handle Teletype Commands

The commands are listed in alphabetical order in Table 1. The pseudo-op ".SIXBi" causes the assembly of the 3 characters between the dots (e.g. "DMP") to be stored as one word stripped ASCII format (e.g. DMP = 041417). This technique was first used in the computer code FLOTSAM⁵ by A. Marusak.

The last ten instructions listed in this section do not belong to the executive routine, and are pertinent only for the present experiment.

```

/EXECUTIVE ROUTINE TO INTERPRET TYPEWRITER MESSAGES
/THE BOOTSTRAP PART OF THE PROGRAM SETS UP A JUMP TO EXECIN FOR THE FIRST
/KEYBOARD INTERRUPT. ANY SUBROUTINE NEEDING KEYBOARD INPUT RESETS THIS JUMP
/UPON COMPLETION OF ITS INPUT.
/EXECIN PACKS INPUT CHARACTERS IN STRIPPED ASCII THREE TO A WORD.
/UPON COMPLETION OF THE INSTRUCTION INPUT BY A CARRIAGE RETURN OR A SPACE,
/EXECIN STORES THE INPUTTED INSTRUCTION AT THE END OF A TABLE OF LEGITIMATE
/INSTRUCTIONS ALSO STORED IN STRIPPED ASCII. THEN IT COMPARES THE INPUT
/INSTRUCTION TO THE LIST JUMPING TO AN ADDRESS IN A CORRESPONDING TABLE,
/IF THAT JUMP HAS NOT BEEN DISABLED.
/FOR JUMPS WHICH CAN BE DISABLED THE WORD STORING THE JUMP ADDRESS HAS
/AC0=0 (NOT JMP). THOSE WHICH CANNOT BE DISABLED HAVE AC0=1(JMP).
/THOSE WHICH CAN BE DISABLED HAVE IN AC1-4 THE NUMBER OF THE BIT IN CODEWD
/WHICH CORRESPONDS TO THE ENABLE-DISABLE KEY. IN CODEWD,
/A 1-BIT ENABLES A 7-BIT DISABLES.
/IF THE INPUT INSTRUCTION AGREES ONLY WITH ITSELF (IT IS STORED AT THE END
/OF THE TABLE) IT EXECUTES AN ERROR JUMP.
EXECIN  DAC EXEC1
        JMS INTYPE
        JMS OCRLF           /QUIT ON CARRIAGE RETURN
        JMP EXEC12
        SAD 0C240           /ALSO ON SPACE
        JMP EXEC12+1
        LRS+6
        LAC EXEC1
        LLS+6
        JMP EXECIN
EXEC12  JMS CRLF
        LAC 0LACEX
        DAC INTYPE
        LAC 0EXEC3
        DAC EXEC2
        LAC EXEC1
        SAD 0EXEC2
        JMP 0+4
        ISZ EXEC2
        ISZ EXEC2
        JMP 0+4
        ISZ EXEC2
        LAC 0EXEC2
        AND 0MEMM4
        DAC EXEC4
        LDI EXEC2
        SPAICLL
        EXIT EXEC7
        LRS+15
        TAD 0AL52           /FIND BIT IN CODE WORD
        DAC 0+2
        LAC CODEWD
        XX

```

```

LOCSPA SPA
EXIT EXECT4
LAW ERMS1-1
ERRT JMS NMSTAS
CRLFDM JMS CRLF
LAC .LACEX
DAC INTYPE
.DISHI JMP DISHIS
QCRLF XX /QUIT ON CARRIAGE RETURN OR LINE FEED
SAD .C215
EXIT QCRLF
SAD .C212
EXIT QCRLF
ISE QCRLF /IF NEITHER, WUMP EXIT
EXIT QCRLF

/STORAGE OF STRIPPED ASCII TYPE-IN INSTRUCTIONS
EXEC13 .SIXBT .DMP. /DUMP RUN ON DEC TAPE
LAC DUMPER
.SIXBT .DIR. /TYPE OUT DEC-TAPE DIRECTORY
LAC DIRTYP
.SIXBT .TYI. /TYPE IN N AT N
LAC TYPINH
.SIXBT .MLT. /STOP DATA ACUMULATION
JMP STOPIT
.SIXBT .STR. /START DATA ACU
XOR= STRIT
.SIXBT .GET. /GET INFO FROM DEC TAPE
LAC GETDTP
.SIXBT .OVR. /TYPE OVERFLOW LIST
JMP OVTYP
.SIXBT .DDT. /JMP TO DDT FOR DEBUGGING
JMP 16000
.SIXBT .INT. /INTEGRATE BETWEEN LINES
DZH= INTEGR
.SIXBT .TOL. /TYPE-OUT BETWEEN LINES
DZH= TOLNS
.SIXBT .KLN. /KILL DISP LINES
JMP LINKIL
.SIXBT .BKK. /BOOKKEEPING AND INITIALIZE
LAC INITIA
.SIXBT .LOL. /SET LOW LINE
JMP ALO
.SIXBT .HIL. /SET HIGH LINE
JPP AHI
.SIXBT .DEC. /SET FOR DECIMAL INPUT
JMP DEC
.SIXBT .OCT. /SET FOR OCTAL INPUT
JMP OCT
.SIXBT .MMA. /SET GAMMA EXPR
LAC GAMA
.SIXBT .ETA. /SET META EXPR
LAC BETA
LOLINE .SIXBT .LOW. /TYPE OUT CHANNEL AND CONTENTS OF LOW LI
DZH= LOQA
HILINE .SIXBT .HIG. /TYPE CH & CONTENTS OF HIGH LINE
DZH= HIEQA
.SIXBT .LAS. /TYPE NUMBERS, CONTENTS BOTH LINES
DZH= LEQ
.SIXBT .XTN. /CHANGE EXTENSION OF FILE IDENTIFICATION
JMP XTEND
.SIXBT .JFE. /EMERY DUMP
LAC EMERY
.SIXBT .SCA. /TYPE OUT SCALERS
DZH= SCIRTP
.SIXBT .RUN. /TYPE OUT CURRENT RUN NUMBER
DZH= RUNUMB
.SIXBT .BUN. /ABORT CLOSED BUNNY DOOR
STOPT2
.SIXBT .SQU. /SOURCE START
SOURCE

```

```

TSEQA  ,SIXBT .TS@.  /TYPE TIME
TSEQ   ,SIXBT .TEN.  /TRY SENDING IO THE MIGHTY PEOPLE BY WAY OF ROBINHOOD'S BARN
LAC SEND1
EXEC1   XX           /COMMAND STORED HERE. IF COMMAND ->ERROR
ERRA    JMP ERRA
ERRA    LAM ERMS2-1
        JMP ERRT
ERMS1   ,SIXBT .LOCKED OUT@.
ERMS2   ,SIXBT .DO WHAT!@.
7A 1-BIT MEANS UNLOCK & 2-BIT MEANS LOCK.
LOCK    XX
        CMA
        AND CODEMD
        DAC CODEMD
        EXIT LOCK
UNLOCK  XX
        LMO
        LAC CODEMD
        OMO
        DAC CODEMD
        EXIT UNLOCK
CODEMD  LAM-2@      /START LOCKED OUT
/CODEMD 3IT        MNEMONIC      JMP TABLE CODE
/13      STR              XOR I
/1@      LAC
/9        TYPE OUTS      DEM I
/0@      DEM

GAMA    LAC OPRLOC
        DAC BGCLK
        LAC .C2@
        DAC .CLOCK
        JMP DISHIS
BETA    LAC .BCLK
        DAC BGCLK

        LAC .C43
        DAC .CLOCK
        JMP DISHIS

```

E. Interrupt Routine

When an interrupt occurs the contents of the AC and MQ are saved and a search is made for the device causing the interrupt. The devices are listed in priority, with clock interrupt having highest priority. After servicing the interrupt, a check is made to see if another interrupt has occurred before returning to the main routine.

```

/INTERRUPT ROUTINE
/WHEN PIC IS ENABLED AN INTERRUPT CAUSES AN EFFECTIVE JMP B.
/LOCATION 1 CONTAINS JMP INTRUP.
INTRUP  DAC INTRAC
        LACD
        DAC INTRMQ      /SAVE AC AND MQ
        CLSF           /SKIP IF CLOCK FLAG=0
        SPP
ICLK    JMP CLOCK
        CAN 21         /SKIP IF CAMAC FLAG
        SKP
        JMP CAMAC
        OTOF          /SKIP IF DECTAPE FLAG
        SKP
        JMP DECTA2
        KSF           /SKIP IF TELETYPE INPUT FLAG
        SKP
        JMP INTYP
        TSF           /SKIP IF TELEPRINTER FLAG=0
        SKP
        JMP TYPDU
        OTEF          /DEC-TAPE ERROR FLAG
        SKP
        JMP DECTA1
        LPCD          /CLEAR LP INT
        RRB           /CLEAR READER FLAG
        PCF           /PUNCH IS ONLY FLAG LEFT! CLEAR IT
DISMIS  IORS
        AND INTRSK    /CHECK IF ANY FLAGS UP
        SZA
        JMP INTRUP+3  /YES SERVICE ANOTHER INTERRUPT
        LAC INTRMQ
        LMQ           /GETS MQ! NEXT GET AC AND RETURN
        LAC INTRAC
        DBR
        ION
        EXIT 0
INTYP   KRB
        EXIT INTYPE
INTYPE  XX
        JMP DISMIS
TYPDU   TCF
        EXIT TYPDU7
TYPDU7  XX
        TLS
        JMP DISMIS

```

F. CAMAC Routines

The CAMAC controller has five main functions: (a) transfer data from the ADC and Tag to the computer on a cycle-stealing (noninterrupt) basis, (b) read and transfer data from 12 scalers, (c) prepare and transmit display coordinates, (d) transmit instructions to pneumatic tube controller and transfer status conditions to the CP, and (e) clear (zero) ADC registers and scalers. The display also involves cycle-stealing from a predetermined buffer region whose first-word-address (FWA) is in loc DISPLC. An interrupt occurs when all of data from the buffer region has been displayed, causing a reset of the FWA, x-axis increment, and scale according to status of buttons and multiple position switches. The subroutine ZAPPER clears the ADC and Scalers. The instructions for control and status of the pneumatic tube start in location PETER. Instructions for movement of a vertical line by push button start in location MARK.

The CAMAC routines are based upon CAMAC organization and modular instrumentation standards.²⁻⁴ A detailed description of CAMAC instructions is given in the appendix, along with examples taken from ORCODE.77

```

/START CAMAC ROUTINES
DSP      HLT
        JMS DOIT
        EXC=B18+F16+DSPH+A2
        LAC ,C1400      /1000 BIT TO ENABLE MARKER INTERRUPT
        TAO XINCR
        JMS DOIT
        EXC=B18+F17+DSPH
        JMP DSP

DOIT     HLT              /"CAN 5" ROUTINE, INTERRUPT OFF
        CAN 64
        LAC DOIT
        CAN 5
        JMP ,=1
        ISZ DOIT
        EXIT DOIT

DCN      HLT              /ENTER WITH 1, 2, OR 3
        AND THREE
        JMS DOIT
        EXC=B18+F18+DCNH
        JMP DCN

```



```

ZAPPER  XX
        LAC THREE
        JMS DISAO
        LAC .CLCAN      /CLEAR ADC AND SCALERS
        CAN 24          /HOPEFULLY DOES NOT AFFECT DISPLAY
        LAC THREE
        JMS OCM
        LAC .ICAN
        CAN 24
        EXIT ZAPPER

ADC      XX
        LAC .C21
        JMS DOIT
        EXC=B10-F17+ADCN+A1
        EXIT ADC

DISAO    XX
        JMS DOIT
        EXC=B10-F21+OCMN
        EXIT DISAO

SETUP    XX          /ENTER WITH -1 OR -3
        DAC THP
        LAC .TABM1
        DAC 14
        LAC= 14
        CAN 24
        LAC= 14
        CAN 44
        ISZ THP
        JMP .-5
        JMP= SETUP

TABLEC   4
        B10-F2+ADCN      /ADC MODULE 14
        5

BUFFER   5200
SIX      6
        -400
        7
        3

EIGHT    12
        B10-F16+DSPN+A3  /DISPLAY (MODULE 13)
        11

DISPLC   56000
        12

DISPLN   -1000
XINCR    10
/CANAC   INTERRUPT COMES HERE
CANAC    CAN 72          /READ REG 3
        DAC SAVCAN      /SAVE IT FOR LATER
        CAN 1           /UPDATES LAM STATUS REGISTER
        JMP .-1
        CAN 32          /READ LAM STATUS REGISTER
        SMA
        JMP MARK
        LAC .CM7        /READ WORD COUNT OVERFLOW STATUS
        CAN 5
        JMP .-1
        CAN 72
        SMA
        JMP .DSPOUT
        JMS PROCES
CANOIS    LAC SAVCAN
        CAN 64
        JMP OISNIS

SAVCAN    0
/MARKER   INTERRUPT COMES HERE
MARK      HAL
        SPA
        JMP PETER
        ALS=4
        SPA
        JMP MARKER

```

```

ASCINT  JMS DOIT
        EXC=B18+F1+ASCHN+A14
        CAM 72
        RAR
        SZL
        JMP ASCKEY
        JMP ASCTEL
ASCKBR  JMP DISHIS
        JMP CANDIS
ASCKEY  JMS DOIT
        EXC=B18+F2+ASCHN+AB
        CAM 72
        AND .C177
        JMP ASCKBR
ASCTEL  JMP DISHIS
        JMS DOIT
        EXC=B18+F16+ASCHN+AB
        JMP CANDIS
PETER  JMS BUNSTA      /BUNNY INTERRUPT WHICH ONE?
        ALS+12
        SHA
        JMP BUN2
        LAC INTIME
        TCA
        DAC 7
        LAC .JENDIN
        DAC ICLK
        CLON          /IN REACTOR INTERRUPT
        LAC .C400      /DISABLE THAT INTERRUPT
        JMS DOIT
        EXC=B18+F21+BUNN
        JMP CANDIS
BUN2    RTL
        SHA
        JMP BUNERR      /?? NOT PROPER INTERRUPT
        LAN 7
        DAC CLOCK4
        LAC .C100
        JMS DOIT      /DISABLE SECOND INTERRUPT
        EXC=B18+F21+BUNN
        JMP CANDIS
BUNERR  LAN BUNMS-1
        JMS NMSTAS
        JMS CRLF
        LAN 777        /CLEAR BUNNY BITS
        JMS DOIT
        EXC=B18+F21+BUNN
        LAC .C200
        JMS UNLOCK
        JMP CANDIS
BUNSTA  XX            /READ RABBIT CONTROL AND CLEAR INTERRUPT
        LAC .CM10
        CAM 9
        JMP :=1
        CAM 72
        EXIT BUNSTA
BUNMS   .SIXBT .RABBIT ERROR.
CAMAC2  XX
        DAC 11
        LOI 11
        CAM 24
        LOI 11
        CAM 44
        LOI 11
        CAM 24
        LOI 11
        CAM 44
        EXIT CAMAC2      /WORD COUNT OVERFLOW SHOULD NOW GO AWAY

```

G. Teletype Input and Output Routines

The TYPEIN routine allows 6 digits to be input, one - sign and/or one decimal point if either needed, followed by E, followed by a + or - sign followed by a one-digit exponent. A comma (,) or carriage return (+) terminates the number. A control-C (+C) returns to the main program. Any other character causes a question mark to be typed and the program returns to the beginning of TYPEIN to try again. The FPTYP routine types a floating-point no. in a F6.1 format. (We do not need more than one digit to the right of the decimal point in the present program.) Integer type out routines are PRINTIN and PRINTDB. Stripped ASCII character type out starts at NMSTAS. Subroutine TYP7BT causes type out of 7-bit ASCII format.

Type in of ASCII information uses subroutine ASCIN. Input text may be terminated by an "␣" sign or carriage return, or by exceeding the allotted number of characters. To recover from an error, the RUBOUT key may be used, or else use control-U (+U) to restart the text input from the beginning.

```

/IN AND OUT TO TELETYPE ROUTINES
/ROUTINES TO TYPE OUT AND READ IN NUMBERS FROM KEYBOARD
DEC    LAM-12+1      /TYPE-IN IN DECIMAL
LOC SKP SKP
OCT    LAM-10+1      /TYPE-IN IN OCTAL
      DAC STOR6
      TCA
      DAC STOR6
      DAC TWELV
      DAC TWELV
      JMP DISMIS
/ROUTINE TO ACCEPT FROM KEYBOARD A DECIMAL OR OCTAL INTEGER
/ REVISED DEC 74 TO ALLOW A FLOATING POINT NUMBER INPUT
TYPEIN XX
      -/
      DAC CQ1        /TILT ON 7TH DIGIT
      DEN ACCN        /ACCUMULATED NUMBER
      DEN EXPCT       /EXPONENT COUNT
      DEN ESIGN       /ZERO, PLUS, NONZERO, MINUS
      LAC OPRLOC      / LAC (OPR
      DAC CHCND
      JMS SPACE1     /TYPE A SPACE
      SKP
      DAC ESIGN
TYPEIN: JML INTYPE
      S40 ,C20J      /CONTROL C, FORGET IT.

```

```

      JMP CALFOM
      SLD .C254
      JMP TEN0+1
      JMS GCRLF
      JMP TEN0
      SLD .C253
      JMP TYPE01
      SLD .C256
      JMP ONTYPE+3
      SLD .C255
      JMP TYPE01+1
      SLD .C305
      JMP TYPE02
      C75 TNUM0
      LAC ACCH
      JMS M0Y12
      TAO C05
      DAC ACCH
      NOP
CHCND  ISE C01
      JMP TYPE01
ONTYPE LAN 27
      JMS TYP0UT
      JMP TYPEIN+1
      LAC .132EX
      DAC CHCND
      JMP TYPE01
      DAC ESIGN
      JMS INTYPE
      SLD .C253
      JMP TYPE02
      SLD .C255
      JMP TYPE02
      JMS TNUM0
      LAC ESIGN
      SLD .C255
      JMP .04
      LAC C05
      TCA
      SKP
      LAC C05
      TAD EXPT
      DAC EXPT
      JMS INTYPE
      JMS GCRLF
      JMP .04
      SLD .C254
      JMP .03
      JMP ONTYPE
      JMS GCRLF
      LAC ACCH
      JMS RFLT.
      LAC EXPT
      DAC EXPON
      SNA
      JMP TYPE03
      SNA
      JMP .010
      JMS XPH.
      LAC STON07
      JMS RFLT.
      JMS XPH.
      ISE EXPT
      JMP .05
      JMP TYPE03
      TCA
      DAC EXPT
      JMS XPH.
      LAC STON07
      JMS RFLT.
      JMS XPH.

```

/COMMA ENDS THE INTEGER
 /ALSO CARRIAGE RETURN
 /IGNORE PLUS SIGN
 / 256=DECIMAL POINT
 / 255=MINUS SIGN
 /NOTE THAT THE MINUS CAN COME ANYTIME
 / 305=1 FOR E-FORMAT
 /TEST FOR NUMBER

/TYPE A 7-MARK
 /BACK TO 60, NO \$200.00

/FOR WHAT FOLLOWS DECIMAL POINT
 /ONE DIGIT EXPONENT ONLY
 / + SIGN
 / -SIGN

/MULTIPLY BY TEN
 /UNLESS EXPONENT REQUIRES DIVID BY TEN

TEN0

```

ISZ EXPCY
JMP ,=5
LAC ZSIGN      /PICK UP SIGN
SZA
JMS XFNEG.
TYPEN3 LAC ,LACEX      /RESET FOR FURTHER KEYBOARD INTERRUPTS
DAC INTYPE
LAC ACCR      /PICK UP INTEREN
EXIT TYPEIN
EXPON
SPACE1 XX      /TYPE A SPACE
LAN 248
JMS TYP0UT
EXIT SPACE1
FPTYP XX      /TYPE A FLOATING POINT NUMBER FROM FLOATING-
/ POINT (3-WORD) ACCUMULATOR TO ONE DECIMAL
/ DIGIT.
JMS XFUNF.
DAC FPT1
JMS PRNTIN
LAC ,C258
JMS TYP0UT
JMS XFH.
LAC FPT1
JMS XFLT.
JMS XFNEG.
JMS XFA.
S2
JMS XFH.
LAC STORG7      /...I.E. MULTIPLY BY TEN
JMS XFLT.
JMS XFH.
JMS XFUNF.
AND ,C17
TAD ,C268
JMS TYP0UT
EXIT FPTYP
TNUMB XX      /CHECK FOR NUMBER
AAC-268
SPA
JMP ONTYPE      /ERROR! ILLEGAL CHARACTER
DAC C05
TAD STORG6
SMA:STL
JMP ONTYPE      /ERROR! ILLEGAL CHARACTER
EXIT TNUMB
MBY12 XX      /MULTIPLY BY IEN
CLL
MUL
STORG7 12      /18 FOR OCTAL
LAC0      /NOTE DONT SAVE HIGH-ORDER
EXIT MBY12
STORG6 LAN-12+1      /LAN-18 1 FOR OCTAL
/CARRIAGE RETURN LINE FEED
CRLF XX
LAN 215
JMS TYP0UT
LAN 212
JMS TYP0UT
EXIT CRLF
/ROUTINE FOR SETTING UP N WORDS AT N
NHACO HLT
LAC ,BNP      /No
JMS STRTYP
CMA
DAC INOUT4
LAC ,BNP      /No
JMS STRTYP
LMO
LAC INOUT4
EXIT NHACO

```

```

/ROUTINE TO TYPE OUT A STRIPPED ASCII WORD AND TO ACCEPT AN INTEGER
STRTYP XX
    JMS STRASC      /STRIPPED ASCII WORD IN AC
    JMS TYPEIN      /TYPE IN A NUMBER
    JMS CRLF
    LAC ACCN        /PICK UP TYPE IN NUMBER
    EXIT STRTYP
/TYPE OUT IN DECIMAL THE INTEGER IN AC
/ALL INTEGER TYPE-OUTS USE THE DOUBLE PRECISION ROUTINE!
/SINGLE PRECISION ROUTINE ZEROS THE MOST SIGNIFICANT WORD.
PRNTSZ XX          /SUPPRESS LEAVING ZEROS
    DAC DOUB1+1
    LAC OPRLOC
    DAC PRNTN4
    CLA:CLL
    JMS PRNT08
    LAC PRNTN2+1
    DAC PRNTN4
    EXIT PRNTSZ
PRNTIN XX
    DAC DOUB1+1      /PUT INTO LOW ORDER BITS
    CLA:CLL          /ZERO HIGH ORDER BITS
    JMS PRNT08
    EXIT PRNTIN
/ROUTINE TO TYPE OUT DOUBLE PRECISION INTEGER
/ENTER WITH LOW ORDER BITS IN DOUB1 1 HIGH ORDER BITS IN AC.
PRNT08 XX
    DAC DOUB1        /HIGH ORDER BITS IN AC
    LAC .INOT1
/CONVERT THE DOUBLE PRECISION OCTAL INTEGER TO A STRING OF DECIMAL DIGITS.
    JMS DBTDEC
    LAM-6
    DAC PRNTT2
    LAC .INOTT        /REF AS PRNTN2+2 IN DBTDEC
    DAC 18
PRNTN3 LAC 18
    SZA
    JMP PRNTN2
    LAC INOTT        /INOTT0 STARTS OFF WITH A 240 (SPACE).
PRNTN4 JMS TYP0UT
    ISZ PRNTT2
    JMP PRNTN3
    LAC 18
/LAST CHARACTER CANNOT BE A SPACE! MUST BE 0-9.
    TAD PRNTN2+2
    JMS TYP0UT
    LAC .C240
    DAC INOTT
    EXIT PRNT08
PRNTN2 TAD 1+2
    JMS TYP0UT
    LAM 160
    DAC 011
    LAC 1
    DAC PRNTN4
    JMP PRNTN3+5
INOTT0 240
    LOC INOTT0+18
/INOTT0 AREA IS USED AS TEMPORARY STORAGE IN IDENT5
/DBTDEC IS A SINGLE PRECISION OCTAL TO DECIMAL CONVERSION ROUTINE.
DBTDEC C
    DAC DOUB1+1      /USED AS INOTT0 18 IN PRNT08
    ZM DOUB1         /AC INTO LOW ORDER BITS
    LAC .INOT6
    JMS DBTDEC
    EXIT DBTDEC
DBTDEC XX
    DAC 9TOET3      /DOUBLE PRECISION BINARY TO
    LAC DOUB1        /DECIMAL ROUTINE
    CLL
    ICIV

```

```

TWELV8 12
        DAC DBTUE1
        LACQ
        DAC DOUB1
        LAC DOUB1+1
        LMO
        LAC DBTUE1
        DIV

TWELV1 12
        DAC DBTUE3
        LACQ
        DAC DOUB1+1

LAMLOC  LAM          /REF AS LAMLOC
        TAD DBTUE3
        SAG .INOTT
        EXIT DBTDEC
        JMP DBTDEC+1

DOUB1   XX          /HIGH ORDER BITS
        XX          /LOW ORDER BITS

/TYPE IN N WORDS AT M
TYPEIN  JMS NMACO
        JMS NMIND
        JMP CRLFOM

/ROUTINE TO ACCEPT N INTEGERS FROM KEYBOARD! STORE BEGINNING AT M.
NMIND   XX
        DAC NMINDM
        JMS LACOM1
        DAC 18
        ISZ NMINDM
        SKP
        EXIT NMIND
        JMS TYPEIN
        DAC 18
        JMP .+5

/TYPE N STRIPPED ASCII WORDS STORED BEGINNING AT M.
/ONE EXITS FROM NMSTAS WHEN STRASC WHICH IS CALLED BY NMSTAS, EXITS WITH A
/ZERO IN THE AC. STRASC EXITS WITH A ZERO IN THE AC IF IT ENCOUNTERS A
/STRIPPED ASCII CHARACTER CONSISTING OF 80.
NMSTAS  XX          /CLEARS LINK AND AC ON EXIT
        AND .MEMMA
        DAC 18
        LAC 18
        JMS STRASC
        SZA:CLL
        JMP .+3
        EXIT NMSTAS

/TYPE S STRIPPED ASCII 3 CHARACTERS PER WORD STORED IN AC ON ENTRY.
STRASC  XX          /CLEARS LINK ON EXIT
        DAC STRAT1
        LA 1-2
        DAC STRAT2
        LAC STRAT1
        CLL
        CLQ
        LRS+14
        OMQ
        DAC STRAT1
        AND .C77
        SNA:CLL
        EXIT STRASC
        SAG .C37
        JMP CRLF37
        AAC-48      /TAD (-37-1)
        SNA
        AAC-188
        AAC 348
        JMS TYPEOUT
        ISZ STRAT2
        JMP STRAS1
        CLC:CLL
        EXIT STRASC

```

```

CRLF37 JMS CRLF
        JMP STRAS3-1
LACOM1 XX
        LACO
        AND ,M777
        AAC-1
        EXIT LACOM1

```

```

/-----

```

```

ASCIN  XX /MESSAGE LOCATION AT ENTRY
        AND ,MEMMA / THEN NUMBER OF ENTRIES ALLOWED
        DAC BUF / THEN INITIAL MESSAGE TO OUTPUT
        AAC-1 / (LOCATION OF)
        DAC BUFST
        TAO* ASCIN
        IAC
        DAC BUFEND
        ISZ ASCIN
ASCIN8 LOI ASCIN
        SZA
        JMS MASTAS
        JMS CRLF
        JMS INTYPE
        JMS QCRLF
        JMP ASCIN*
        SAO ,C300 /300 IS 0 TERMINATES MESSAGE
        JMP EQMA
        ISZ ASCIN
        DAC TALLY
        LAN ASCIN1
        AND ,MEMMA
        DAC INTYPE
        JMS D22BUF
        LAC TALLY
ASCIN1 SAO ,C300
        JMP EQMA
        SAO ,C377
        JMP RUBOUT
        SAO ,C203
        JMP CRLFON
        JMS QCRLF
        JMP EQMA
        SAO ,C225 /FOR 0U TO NEW ENTIRE LINE
        JMP UPU
        STI BUF
        ISZ BUF
        LAC BUF
        SAO BUFEND
        JMP EQM8
        JMS D22BUF
ASCIN2 JMS INTYPE
        JMP ASCIN1
EQMA ISZ BUF
        LAC BUF
        SAO BUFEND
        JMP EQM8
        JMS D22BUF
        JMP EQMA
EQM8 LAC ,LACEX
        DAC INTYPE
        EXIT ASCIN
UPU JMS CRLF
    4
        TAO ASCIN
        DAC ASCIN
        EXIT ASCIN
RUBOUT LAC BUF

```



```

SAD BUFST
JMP ASCIN2
AAC -1
OAC BUF
LAC BUF
JMS TYP7BT
JMP ASCIN2
DE2BUF XX
LAC .C100
STI BUF
EXIT DE2BUF

TMYB 0
FSPECT 0
SCLOC1 0
MHCMDM 0
OVCTR 0
BUF 0 /ALSO USED IN TYP7BT
BUFST 0 / DITTO
BUPEND 0
TALLY 0 /ALSO USED IN BEGIN, LOGFIX AND TYP7BT
VAL1 0 /TYPE OUT CONTENTS OF AC
TLOC 0 /TORDER ROUTINE
TLOC1 0 / DITTO
SCLOC2 0 /SCALER TYPE OUT
TMR 0 /SETUP ROUTINE
OVc3 0 /TYPE OUT OF OVERFLOW
OVc4 0 / DITTO

TYP7BT XX /TYPE OUT OF 7-BIT FORMAT STUFF
OAC TALLY / COUNTER IN AC AT ENTRY
LACQ / LOCATION OF INFORMATION -1 IN HQ
OAC 13
LDI 13
OAC BUFST
JMS LLS7
JMS LLS7
LAC BUFST
LMO
ZAP
LLS-4
OAC BUF
LDI 13
LMO
LAC BUF
LLS-3
OAC BUF
LACQ
OAC BUFST
LAC BUF
SAD .C100 /100 ENDS MESSAGE
JMP .+6
JMS TYP7BT
JMS LLS7
JMS LLS7
ISZ TALLY
JMP TYP7BT+4
JMS CRLF
EXIT TYP7BT

LLS7 XX
LAC BUFST
LMO
ZAP
LLS-7
OAC BUF
LACQ
OAC BUFST
LAC BUF
SAD .C100
JMP LLS7-2
JMS TYP7BT
EXIT LLS7

```

H. Type Out Contents of Channel in AC

A particular example of the use of type-out routines, both text and integer, are given in this short section.

```

/TYPE OUT CONTENTS OF CHANNEL IN AC.
VAL      XX
        TAO REG1
        DAC /AL1
        LAC* VAL1
        JMS PRNTIN
        EXIT VAL
LOEQ     XX      /TYPE OUT CHANNEL AND CONTENTS OF LOW LINE
        LAC LINELO
        JMS PRNTSZ
        LAC LINELO
        JMS VAL
        EXIT LOEQ
HIEQ     XX      /TYPE OUT CHANNEL AND CONTENTS OF HIGH LINE
        LAC LINEHI
        JMS PRNTSZ
        LAC LINEHI
        JMS VAL
        EXIT HIEQ
LEQ      LAC LOLINE
        JMS STRASC      /TYPE OUT LOW AND HIGH LINES
        JMS LOEQ
        JMS CRLF
        LAC HILINE
        JMS STRASC
HIEQA    JMS HIEQ      /HIGH LINE ONLY
        JMP CRLFDM
LOEQA    JMS LOEQ      /LOW LINE ONLY
        JMP CRLFDM

```

I. Scaler Routines

The first routine is the CAMAC-controlled scaler read for 12 scalers.

These data are stored starting in loc SCLOC1.

Once the scaler data are in storage, the values may be typed out using routine SCLRTP.

```

SCALAR  XX          /SCALER READ IN FROM CAMAC
        LAC SCLOC1  / FOR TWELVE SCALERS
        DAC 13
        -14
        DAC SCL1
        LAC ,ASCNS
        CAM 5
        JMP ,=1
SCA1    LAC ,EXC
        CAM 7
        JMP ,=1
        CAM 1
        JMP ,=1
        CAM 72
        DAC 13
        CAM 72
        DAC 13
        ISZ SCL1
        JMP SCA1
        EXIT SCALAR

SCLRTP  JMS HGET
        SKP
        JMP ERRA
        TAD ,TAB1
        DAC SLOC2
        LAM
        TAD SLOC2
        DAC SLOC2
        JMS SCATYP
        JMP DISMIS
NGET    XX
        LAC ,BNP
        JMS STRTYP
        SPA:5NA
        JMP ERRA
        DAC ,MINON
        TCA
        TAD ,NUMSP

```

SPA
ISZ MGET
LAC MMINON
EXIT MGET

SCATYP

XX
LAC SLOC2
DAC 14
-14
DAC SCL2
-6

DAC SCLB
JMS SPACE1
LAC 14
DAC DOUB1+1
LAC 14

SCAT1

JMS PRINTB
ISZ SCLB
SKP

JMS CRLF
ISZ SCL2
JMP SCAT1
JMS CRLF

EXIT SCATYP
2

SCLB
SCL1
SCL2
/

XX
XX

J. Start and End of Run Routines

At the STR command the program goes to STRTIT. Spectral regions, ADC, scalers, overflow lists are all set to zero, CAMAC controller locations are reset, and the pneumatic tube status is read in. If all okay, the CAMAC interrupts are enabled, the clock is turned on, and the rabbit movement is initiated if called for. At the end of the run the program goes to STOPIT.

```

-----
SETCLK  XX          /RESET CLOCK
        LAM-7401    / ONE SECOND
        DAC 7
        EXIT SETCLK
STOPIT  CLA:IAC      /(IAC ONE)
        JMS DISAO
        CLOF
        JMS CLK2
        JMS PROCES
        LAM EGRMS-1
        JMS NMSTAS
        LAC RUNNUM
        JMS PRNTIN
        JMS CRLF
        LAC TSEQA
        JMS STRASC
        LAC 6
        JMS PRNTIN
        LAC .C200    /UNLOCK TYPING COMMAND
        JMS UNLOCK
STOPIT2 LAM 767      /OR JMP CRLF0M FOR SOURCE COUNT
        JMS DOIT
        EXC+B10+F21+0UNN /CLEAN BUNNY BITS
        JMP CRLF0M
EGRMS   .SIXBT .END OF RUN 0.
STRTIT  LAC C200     /START DATA TAKING
        CLL          /ZERO SPECTRAL REGION
        MUL
SPTW    SPECS1+SPECS2
        LAC0
        TCA

```

```

DAC TY
LAC ASCIN
DAC 14
DZM 14
ISE TY
JMP 1-2
JMS 9GIN
-3
JMS SETUP
JMS ZAPPER
JMS ADC
DZM WHCHOM
JMS DZMOVF
CLA
SAD DPENOR /TEST FOR RABBIT OR SOURCE
JMP STRT1
JMS BUNSTA /RABBIT CONTRUL; CHECK HUTCH 1 DOOR
AND .C20
SZA
JMP INITI3
LAC .C511 /ENABLE INTERRUPTS; SHOOT RABBIT
JMS DOIT / TO REACTOR
EXC-B10-F10-BUNN
JMP STRT2
STRT1 LAC STOPT2+3 /FOR SOURCE MEASUREMENTS
DAC STOPT2 / I.E. NO RABBIT!
CLOH
STRT2 LAC .C220 /LOCKOUT STAR! AND TYPING COMMANDS
JMS LOCK
JMP DISMIS

```

K. DECTAPE Routines

These routines replace the Digital Equipment DECTAPE handler. The programming is similar but more compact. One useful feature is the option to recover a DECTAPE read or write command when DECTAPE unit is not ready. After the error message, reset the unit correctly and type control-R (↑R) to recover.

Digital Equipment Corporation's DECTAPE handler provides similar operation error recovery. The difference here is that instead of a single error code (number) saying you did something wrong, in the present code messages are typed indicating what is actually wrong.

```

/-----
/
/ DECTAPE ROUTINES
SETUNT XX
      DZH DTUFLG
      LAW UNIM=1      /SET DEC-TAPE UNIT NUMBER
      JMS NMSTAS
      JMS TYPEIN
      DAC UNITOT
      EXIT SETUNT
UNITOT 7      /DEFAULT=UNIT
DECTAP XX
      JMS DECTPP
      JMP DISMIS
DECTPP XX
      KOR DTREV
      DAC DTTEM
      DTRA
      LRS=6
      AND .C777
      KOR DTTEM
      ALS=6
      DTXA
      DAC DTUFLG
      EXIT DECTPP
DECTA1 DTRB
      DAC DTERR
      JMS DTSTOP
      SKP
DECTA2 ISZ DECTAP
      DZH DTUFLG
      EXIT DECTAP

```

```

DTWRIT XX
      XCT= DTWRIT
      ISZ DTWRIT
      SPA
      JMP DTWRIA
      DAC DTUNIT
      LAC= DTWRIT
      ISZ DTWRIT
      DAC 16
      LAC .DTDA1
      DAC 12
      DZH DTDFLG
      LAC DTWCON
      DAC DTBLK
      JMS DTGJIT
      JMS DTMAP
      JMP DTTFLL
      SPA
      CMA
      DAC DTDATA=377
      -377
      DAC DTGNTA
      DAC DTGNT
      LAC= DTWRIT
      ISZ DTWRIT
      SMAICLL
      JMP DTWRIX
      DAC= 12
      DAC DTCKSH
      SAG THOUS5
      JMP DTWRIB
      DZH= 12
      LRS=19
      AND =C377
      AAC =2
      TCA
      DAC DTGNT
      -375
      DAC DTGNTA
      JMS DTBLOC
      DAC DTDATA=1
      SKP
DTWRIX JMS DTBLOC
      LAC DTGNTA
      SMA
      JMP =+4
      DZH= 12
      ISZ DTGNTA
      JMP =+2
      LAM 244
      JMS DTGO
      JMS DTWERR
DTWRI1 ISZ DTWRIT
      EXIT DTWRIT
DTWRIA CMA
      DAC DTUNIT
      JMS DTWRI2
      JMP DTWRI1
DTWRIB TCA
      DAC= 12
      ISZ DTGNT
      DZH= 12
      ISZ DTGNT
      JMP =+1
      LAM
      DAC DTDATA=377
      LAM 244
      JMS DTGO
      JMS DTWERR
      JMS DTGOIR

```

/DECTAPE FULL


```

LAC ,DTDIR
DAC DTTEMA
LAC ,DTOAT
DAC DTTEMB
-48
DAC DTCNT
LAC= DTTEMB
XOR= DTTEMA
DAC= DTTEMA
XOR= DTTEMB
DAC= DTTEMB
ISZ DTTEMA
ISZ DTTEMB
ISZ DTCNT
JMP ,+18
-3
DAC DTCNT
LAC DTWFIL
DAC 16
LAC DTWFIL+2
DAC 12
LAC= 16
DAC= 12
ISZ DTCNT
JMP ,+3
LAC DTWFIL+1
DAC= 12
OZH DTDFLG
LAW 244
JMS DTGO
JMS DTWERR
LAC ,DTDA1
CMA
TAD DTWFIL+2
AAC-37
CLL
LRS+5
AAC 71
DAC DTBLK
CLA:CLL
LLS+3
DAC DTWFIL+2
LAW 224
JMS DTGO
JMS DTWERR
LAC DTWFIL+2
SNA:CLL
JMP ,+6
MUL
48
LACO
TAD ,DTDA1
DAC 12
LAC ,DTDI1
DAC 16
-48
DAC DTCNT
JMS DTOLOC
LAW 244
JMS DTGO
JMS DTWERR
JMP DTWRI1
DTREAD XX
XCT= DTREAD
ISZ DTREAD
SPA
JMP DTREAC
DAC DTUNIT
OZH DTDFLG
LAC DTEOF

```

```

SPA
JMP DTREA7
LAC DTRLBK
DAC DTBLK
LAW 224
JMS DTGO
JMS DTRERR
LAC DTRLBK
SPA:CLL
CHA
TCA
TAD DTDATA+377
LAC DTDATA+377
DAC DTEOF
SNL
CHA
DAC DTRLBK
LAC DTREAD
ISZ DTREAD
DAC 12
LAC DTDA1
DAC 16
-377
DAC DTCTNT
LAC DTREAD
SNA:CLL
JMP +10
LRS+10
TCA
DAC DTCTNT
ISZ 16
ISZ 16
ISZ DTCTNT
ISZ DTCTNT
JMS DTBLOC
DTREAB ISZ DTREAD
ISZ DTREAD
LAC DTEOF
EXIT DTREAD
DTREA1 CHA
DAC DTUNIT
DZM DTEOF
LAC DTREAD
ISZ DTREAD
JMS DTDIRS
OPRLOC NOP
SNA
JMP DTREA2
AAC 4
DAC DTTEMB
LAC ,MXCT
XOR DTTEMB
DAC DTRLBK
JMP DTREA7+1
DTREA2 LAW DTREAG+1
JMS MHSTAS
JMP OPRCUT
DTREAG ,SIXBT "NO SUCH FILE"<37><0>
DTEOF 0
DTTYDI XX
LAC DTTYDI
DAC DTUNIT
ISZ DTTYDI
JMS DTGOIR
-70
DAC DTCTNT
LAC ,DT40
DAC 16

```

```

DTTYOA  LAC= 16
        JMS STRASC
        SNA
        JMP DTTYOC
        LAC= 16
        JMS STRASC
        LAN 248
        JMS TYP0UT
        LAC= 16
        JMS STRASC
        JMS CRLF
DTTYDB  ISZ 16
        ISZ DTENT
        JMP DTTYOA
        EXIT DTTYDI
DTTYOC  ISZ 16
        ISZ 16
        JMP DTTYDB
DTSEAR  KX
DTRTN   LAC DTBLK
        SPA:CLL
        STL:CHA
        DAC DTCLK
        DAC DTCLK+1
        SZL:CLA
        LAC ,C400
        DAC DTREV
        SZL:CLL:CLA
        IAC
        TAD ,DTCL
        DAC 31
        D2M 30
        LAN 216
        JMS DECTAP
        JMP DTENDZ
DTRTY   LAC DTCLK
        AND ,C7777
        SAD DTCLK+1
        EXIT DTSEAR
        CLC
        TAD DTCLK+1
        CHA
        TAD DTCLK
        SNA
        JMP DTURN
        DAC 30
        LAN 214
        JMS DECTAP
        JMP DTENDZ
        JMP DTRTY
DTURN   IAC
        CHA
DTURN1  DAC 30
        LAN 714
        JMS DECTAP
        NOP
        LAN 214
        JMP DTRTY-2
DTENDZ  LAC DTERR
        RTL
        SNA
        JMP ,+3
        =2
        JMP DTURN1
        LAN DTUNMG=1
        JMS DTUSTA
        JMS DTUNST
        JMP DTRTN

```

```

DTSTOP XX
      OZM DTUFLC
      LAC .0204
      OTXA
      EXIT DTSTOP
DTCOLK 0
      0
DTAEV  0
DTERR  0
DTOLK  0
DTTEH  0
OTCNT  0
DTUNIT 0
DTRLBK 0
DTUCBK 0
DTUSTA XX
      DAC DTUST1
      LAC DTERR
      ALS-3
      SPA
      JMP .-3
      JMS DTBAD
      EXIT DTUSTA
DTUST1 LAM DTUMMG-1
      JMS DTWAIT
      EXIT DTUSTA
DTWERR XX
      LAM DTWRMG-1
      JMS DTUSTA
      -3
      TAD DTWERR
      DAC DTWERR
      EXIT DTWERR
DTRERR XX
      LAM DTUMMG-1
      JMS DTUSTA
      -3
      TAD DTRERR
      DAC DTRERR
      EXIT DTRERR
DTUMMG .SIXBT "UNIT NOT READY"<37><3>
DTWRMG .SIXBT "WRITE LOCKOUT"<37><3>
DTWAIT XX
      JMS MHSTAS
      LAM DTDFXM-1
      JMS MHSTAS
      JMS INTYPE
      LAC .LACEX
      DAC INTYPE
      KRB
      SAG .C222
      EXIT DTWAIT
      JMP CPROUT
-DTOFXM .SIXBT "FIX AND TYPE ,R FOR RSTY, ANY KEY TO ABORT"<37><3>
DTBAD  XX
      LAC DTERR
      XX
      EXIT DTBAD
DTUNST 0
      LAC DTERR
      SPA:CLL
      JMP .-5
      OTRA
      LRS-17
      SAG DTUNIT
      EXIT DTUNST
      CLA
      JMS OCTPP
      OZM DTERR
      LAC DTUNIT
      ALS-17
      OTLA
      EXIT DTUNST

```

```

DELTA=5
DTDIRS  XX
        DAC DTTEMB
        LAC DTDFLG
        SAD DTUNIT
        SKF
        JMS DTGDIR
        LAC DT48
        DAC DTTEMA
        DAC 12
        -78
        DAC DTENT
        DZH DTDF1
        DZH DTDF2
DTDIR1  LAC DTTEMA
        DAC 16
        -3
        DAC DTENTA
DTDIR2  LAC 12
        SNA
        JMP DTDIR4
        SAD 16
        JMP DTDIR3
DTDIR3  LAC DTTEMA
        AAC 4
        DAC DTTEMA
        DAC 12
        ISZ DTENT
        JMP DTDIR1
        LAC DTDF1
        SZA
        ISZ DTDIRS
        LAC DTDF2
        EXIT DTDIRS
DTDIR3  ISZ DTENTA
        JMP DTDIR2
        LAC DTTEMA
        DAC DTDF2
        JMP DTDIR5
DTDIR4  LAC DTDF1
        SZA
        JMP DTDIR5
        LAC DTTEMA
        DAC DTDF1
        JMP DTDIR5
DTGDIR  XX
        LAC DTUNIT
        DAC DTDFLG
        LAC C188
        DAC DTSLK
        LAH 224
        JMS DTG0
        JMS DTRERR
        EXIT DTGDIR
DTG0    XX
        DAC C18
        JMS DTURBY
        JMS DTUNST
        JMS DTSEAR
        LAC DTDA1
        DAC 31
        -8774
        DAC 38
        XX
        JMS DECTAP
        SKF
        ISZ DTG0

```

```

-3
DAC 30
LAC .OTCBL
DAC 31
LAW 214
JMS DECTPP
LAC .OTUTS
DAC DECTAP
EXIT DTGO
DTUBSY XX
LAC DTUFLG
SNA:CLC
EXIT DTUBSY
DAC DTREG
JMP DISNIS
DTUTST OPR
LAC DTREG
SNA
JMP .+3
OEM DTREG
EXIT DTUBSY
JMS DTSTOP
JMP DISNIS
DTUFLG 0
DTREG 0
DTCNTA 0
DTTEHA 0
DTTEHB 0
OTCIF1 0
OTDIF2 0
OTCFLG 0
OTWRIZ XX
LAC= DTWRIT
ISE DTWRIT
DAC CTWFIL
JMS OTDIRS
JMP OTDFLL
SZA
JMP OTFIAL
LAC OTDIF1
DAC CTWFIL+2
-4E
DAC DTWCT
DAC DTCHT
LAC .DTDA1
DAC 10
LAC .OTD11
DAC 12
DAC OTMTRX
ISE OTMTRX
JMS OTBLOC
OEM OTTEHC
OEM OTMCOX
-1120
DAC OTBLOK
-1
DAC DTREM
CLA:IAC
DAC OTDIRA
JMS OTMPS
JMS OTMAP
JMP OTTFLL
XOR ,HXCT
DAC OTWFIL+1
EXIT DTWRIZ
LAW OTDIFL+1
SKP
OTTFLL LAW OTDTFL+1
SKP
OTFIAL LAW OTDFAL+1
JMS NNSTAS
JMP OPROUT

```

```

/DIRECTORY FULL
/FILE ALREADY EXISTS

```

```

/DECTAP FULL

```

```

DTDFL .SIXBT "DIRECTORY FULL"<37><8>
DTDTFL .SIXBT "TAPE FULL"<37><8>
DTDFAL .SIXBT "FILE ALREADY EXISTS"<37><8>
DTMFIL
      0
      0
      2
DTBLOC XX
      LAC= 16
      DAC= 12
      TAO DTCKSH
      DAC DTCKSH
      ISZ DTGNTA
      NOP
      ISZ DTGNT
      JMP DTBLOC+1
      CMA
      IAC
      EXIT DTBLOC
DTCKSH
DTCSIT
      XX
      LAC DTTBIT
      XOR= DTHTRX
      DAC= DTHTRX
      EXIT DTCSIT
DTHPS
      XX
      =22
      DAC DTRBIT
      LAC= DTHTRX
      CMA:STL
      DAC DTSBIT
      LAC DTDIRA
      SMA:CLA:RAL
      RTR
      DAC DTTBIT
      EXIT DTHPS
DTHPP
DTHPP1
      XX
      LAC DTDIRA
      TAO DTHTRX
      DAC DTHTRX
      JMS DTHPS
      ISZ DTHMCT
      EXIT DTHPP
      =1100
      DAC DTBLOK
      CLA
      ISZ DTDIRA
      =2
      IAC
      DAC DTDIRA
      D2H DTGNT
      TAO DTYENC
      DAC DTYENC
      LAC DTDIRA
      RAL
      LAC CTNCSK
      SNL:CHA
      SKP
      TAO .C1100
      CMA:SHA
      DAC DTREN
      =41
      DAC DTHMCT
      JMP DTHPP1
DTHAP
      XX
      D2H DTGNT
DTHAP1
      LAC DTTBIT
      AND DTSBIT
      SZA:CLC
      JMP DTREN
      ISZ DTBLOK
      JMP DTHAP4
      DAC DTBLOK
      EXIT DTHAP

```

```

DTHAP4  ISZ DTCNT
        LAC DTDIRA
        TAD OTTEMC
        DAC OTTEMC
        LAC DTDIRA
        RAL
        LAC OTTBIT
        SNL:CLL:RAL
        RTR
        DAC OTTBIT
        ISZ DTRBIT
        JMP DTHAP4+1
        JMS DTHPP
        JMP DTHAP1
DTREM   XX
        TAD DTCNT
        SPA:CLC
        JMP DTHAP4
        -DELTA
        DAC DTREM
        LAC DTDIRA
        RAL
        LAC OTTEMC
        SZL:CLL
        CMA
        DAC DTMCBK
        ISZ DTHAP
        EXIT DTHAP
DTTEMC  0
DTHTRX  0
DTRBIT  0
DTMCT   0
DTDIRA  0
DTTBIT  0
DTRBIT  0
DTBLOK  0
UNIN    0
/        .SIXBT .DECTAPE UNIT 0.

```


L. Data Manipulation; Integration and Partial Type Out

The data displayed between the vertical lines may be integrated (summed) using the INT command. One word of caution - the INT command will interfere with data accumulation so should not be used to sum data being accumulated. The centroid is also calculated and typed out.

The data displayed between the vertical lines may be typed out starting at the location specified by the low line, eight channels to a line. After the data have been typed out the integration routine is entered, and the area and centroid are calculated and typed out.

```

/-----
/
SETLOW  XX          /USED IN INTEGRATION AND TYPE OUT LINES
        JMS LNOCHK
        SNA
        JMP CRLFOM
        AAC 2
        TCA
        DAC IG1
        DAC IG5
        DZH IG6
        DZH CENT
        DZH CENT+1
        LAC ,BNE
        JMS STRASC
        LAC REG3
        IAC
        JMS PRNTS2
        LAC REG1
        TAD LINELO
        DAC IG2
        EXIT SETLOW
CENT     0          /LOCATION OF CENTROID VALUE
        8          / IN FLOATING POINT
INTEGR   JMS INTGRA /INTEGRATE BETWEEN LINES
        JMP LEO     /THEN TYPE OUT LINES INTEGRATED BETWEEN
INTEGRA  XX
        JMS SETLOW
        DZH DOUB1
        DZH DOUB1+1
INTR1    CLL
        LAC DOUB1+1
        TAD IG2
        DAC DOUB1+1
        S2L
        ISB DOUB1
        LDI IG2
        JMS XFLT.
        JMS XFW.
        LAC IG6
        SNA
        JMP INTR2
        JMS XFLT.
        JMS XFW.
        JMS XFW.

```

/CALCULATION OF CENTROID

```

      LAC .CENT
      JMS KGET.
      JMS KFA.
      32
      LAC .CENT
      JMS KSTOR.
INTR2  ISZ IG0
      ISZ IG2
      ISZ IG1
      JMP INTR1
      LAC DOUB1
      OAC DATSUM+1
      LAC DOUB1+1
      OAC DATSUM
      LAC .BAE      / A=
      JMS STRASC
      LAC DOUB1
      JMS PRNTOB
      JMS SPACE1
      LAC .CAE      / C=
      JMS STRASC
      JMS KFH.
      LAC .DATSU
      JMS SFLT24
      JMS SFD.
      JMS KFH.
      LAC LINELO
      JMS SFLT.
      JMS KFA.
      32
      JMS PPTYP
      JMS CRLF
      EXIT INTGRA
TOLN5  JMS SETLOW      /TYPE OUT BETWEEN LINES
      JMS CRLF
      JMP TOLN2
TOLN6  -10
      OAC IG3
      LAC IG2
      AND .OMASK
      JMS PRNTIN
TOLN1  LDI IG2
      JMS PRNTIN
      ISZ IG2
      ISZ IG3
      JMP TOLN1
TOLN2  JMS CRLF
      LAC IG1
      AAC 10
      OAC IG1
      SPA
      JMP TOLN6
      AAC-10
      SPA
      JMP TOLN6+1
      JMS INTGRA
      JMP DISMIS

```

M. DECTAPE Directory

Following the command DIR, the tape unit is specified, and the directory of the tape on that unit is typed out, one filename and extension to a line.

```

DIRTYP  JMS SETUNT      /TYPE-OUT OF CONTENTS OF DEC TAPE
        DAG ,+3        / NOT NECESSARILY THE DATA-TAPE UNIT
        JMS CRLF
        JMS DTTYDI
        ?
        JMP DISMIS
RUNUMB  LAM MES2
        JMS NHSTAS
        LAG RUNNUM
        JMS PRNTSE
        JMP CRLFON
LSV     0
RUNNUM  0
STRUM   XX            /GET RUN NUMBER
        JMS CRLF
        LAM MES2-1
        JMS NHSTAS
        JMS TYPEIN
        DAG RUNNUM
        SPA 18NA
        JMP STRUM+1
        JMS IDENTIS
        LAG EXTENS
        JMS STRASC
        JMS CRLF
        EXIT STRUM
        .LST

```

N. Initializing and Bookkeeping Routines

Following the command BKK, the CP types out and receives information as discussed above for Fig. 2. The following portion of the code performs these tasks, including checks of pneumatic tube controller status.

/NEXT ARE THE INITIALIZING AND BOOK-KEEPING ROUTINES

```

INITIA  LAC .DATLO
        JMS ASCIN
        30
        LAN MES1-1
        -10
        JMS LOGFIX
        JMS STRUM
INITI0  LAN MES3-1
        JMS MNTAB
        JMS TYPEIN
        CLL
        MUL
        74
        LAC0
        OAC INTIME
        LAC EXPON
        SNA
        JMP .+12
        MUL
        12
        LAC0
        OAC .+3
        LAC INTIME
        IDIV
        XX
        LAC0
        OAC INTIME
        LAC INTIME
        SPA:0NA
        JMP INITI0
INITI1  JMS ORLF
        LAN MES4-1
        JMS MNTAB
        JMS TYPEIN
        SPA:0NA
        JMP INITI1
        OAC CCT0
        IAC
        CNA
        OAC CCT1
        TAD .NUMS1
        SPA
        JMP INITI1
        JMS ORLF
        LAC .TINTA
        OAC TINTB
        LMO
        LAC CCT1
        JMS MNTAB
        ISS CCT1
        JMS TORDER
        LAC OPRLOC
        OAC INISNT

```

```

INIT0  LAC .SLOC1
      OAC PSPECT
      AAC-1
      OAC ASCIN
      LAN
      DAC FRSTP
      DAC OPENDR
      LAC OPRLC
      OAC CLOC8
      LAC BUN2*2
      OAC CLOC4      /FOR BUNNY ERROR
      LAC .SCLN1
      OAC SCLOC1
INISNT  OPR
      JMS CRLF
      LAN MES6-1
      JMS NMSTAS
      JMS INTYPE
      SAG .C323      /323*8(SOURCE)
      JMP INITI4
      SAG .C322      /322*8(ABBIT)
      STP
      JMP .-18      /NEITHER S NOR M MUST BE ONE
      JMS CRLF
.LAC4   LAC .C14      /LAC 14
      JMS DDIT      /CLOSE MATCH 7 DOOR
      EXC*218*F18*BUNN
      LAN 767
      DAC STOPT2
      LAC SCCLK
      DAC CLOC8
      JMS BUNSTA      /CHECK DOOR 1 STATUS
      AND .C28
      SZA
INITI3  JMS TTNESS
      LAC .C28      /UNLOCK START
      JMS UNLOCK
      JMP CRLFON
INITI4  DEN OPENDR      /OPENDR SET = ZERO FOR SOURCE
      JMP INITI3*1
TTNESS  HLT
      LAN MES5-1
      JMS NMSTAS
      EXIT TTNESS
BOINC   LAC .JINITA
      OAC INISNT
      LAC .TINTA
      OAC INTB
      AAC-1
      OAC 18
      CLAIAC
      DAC 18
      DAC SCT8
      LAC T828A
      JMS STRITP
      IAC
      DAC 18
      -2
      OAC CCT1
      JMP INIT0
INITA   DEN OPENDR
      JMP STRITP
BGCLK   OPR
BGIN     XX
      CLOP
      DEN 6
      JMS SETCLK
      LAC .BUNF1
      OAC BUFFER
      LAC .BUSE
      TCA
      OAC BUFFER*2
      EXIT BGIN

```

```

DZHOVF XX          /ZERO OVERFLOW LIST
      LAC .OVERH
      OAC OVC
      LAC .OVDAT
      AAC-1
      DAC 14
      DZH 14
      ISZ OVC
      JMP -2
      DZH OVCTR
      EXIT DZHOVF
MES1  .SIXBT .INPUT DATA LOG.
MES2  .SIXBT .RUN NUMBER 0 0.
MES3  .SIXBT .IRRADIATION TIME 0 0.
MES4  .SIXBT .NO OF TIME INTERVALS 0 0.
MES5  .SIXBT .SHUT DOOR 10.
MES6  .SIXBT .SOURCE OR READBIT 0.
TORDER XX          /TEST ORDER OF INPUT TIMES
      LAC CCT1
      OAC NMNOM
      LAN
      TAD TMTB
      DAC 12
      CLAIAC        /FIRST TIME VALUE MUST BE .GE.2
      DAC TL0C
TORD1  LDI 12
      DAC TL0C1
      TCA
      TAD TL0C
      SMA
      JMP TORD2
      LAC TL0C1
      DAC TL0C
      ISZ NMNOM
      JMP TORD1
      EXIT TORDER
TORD2  LAN THSC-1
      JMS NMSTAB
      JMS CRLF
      JMP INIT11
THSC  .SIXBT .DATA ERROR.

```

0. Filename and Extension Routines

The identification number (6 characters) are placed in stripped ASCII in locations IDNT and IDNT+1. The extension is stored in IDNT+2 and remains the same until changed using the command XTN.

```

IDENTS  XX
        LAC RUNNUM
        JMS BYDEC
        LAC ,INOTT
        JMS IONTPC
        EXIT IDENTIS
IDENTPC  XX
        DAC 14
        LAC ,IDNT
        DAC LY1
        JMS IDS
        JMS IDS
        JMS IDS
        ISZ LY1
        OZM, LY1
        JMS IDS
        JMS IDS
        JMS IDS
        EXIT IONTPC
IDNT     4E4848
        4E4848
EXTENS   .SIXTY .NO.
XTEND    JMS SPACE1      /CHANGE EXTENSION
        LAC EXTENS
        JMS STRASC
        LAN XMSG-1
        JMS NMSTAS
        LAC ,INOTT
        DAC 14
        LAC ,IDNT
        TAD TWO
        DAC LY1
        JMS INTYPE
        SAD ,C248        / 248=ZERO
        JMP CRLFOH
        JMS OCRLF
        JMP CRLFOH
        SAD ,C248        / 248=SPACE
        JMP CRLFOH
        STI 14           /IF IT GETS TO HERE ITS THE FIRST CHARACTER
                        / OF THE NEW EXTENSION
        JMS INTYPE
        JMS OCRLF
        JMP ZERO
        STI 14           /20 CHARACTER OF EXTENSION
        JMS INTYPE
        JMS OCRLF
        JMP ZERO+1
        STI 14
        LAC ,INOTT
        DAC 14
        JMS IDS
        JMS IDS
        JMS IDS
        JMP CRLFOH
ZERO     OZM, 14
        OZM, 14
        JMP ,+10

```

```

XMSC      .SIXBT " IF ON TYPE 2, ELSE NEW EXTENSION"<37><8>
ID3       XX
          LOI 14
          AAC-12      /LETTERS AND NUMBERS TO SIXBT FORMAT
          SPA         /TO DIFFERENTIATE BETWEEN INOTTD DATA
          AAC-6p      / AND JFE DATA
          AAC-12
          LRS-6
          LOI LY1
          LLS-6
          STI LY1
          EXIT ID3
LRS7      XX
          LOI 14
          LRS-7
          LOI LY1
          EXIT LRS7
LY1       E
LOGFIX    XX
          DAC TALLY
          LAC ,DATL1
          DAC 14
          LAC ,LOGOU
          DAC LY1
LYB       DZH- LY1
          JMS LRS7
          LLS-7
          STI LY1
          JMS LRS7
          LLS-7
          STI LY1
          JMS LRS7
          LLS-4
          STI LY1
          ISZ LY1
          CLA
          LLS-3
          STI LY1
          JMS LRS7
          LLS-7
          STI LY1
          JMS LRS7
          LLS-7
          CLLRAL
          STI LY1
          ISZ LY1
          ISZ TALLY
          JMP LYB
          EXIT LOGFIX

```


P. Clock Interrupt

There are five important clock interrupts. These are (a) at the end of the irradiation period, (b) the time to expose the rabbit in Hood No. 2 to the detector, (c) the beginning of the first counting period, (d) at the end of each counting period, and (e) at the end of the last counting period. The running time in sec is stored in location 6. Location 7 is incremented every 1/60 sec, and clock interrupt occurs when location 7 has a zero in it. This location is set at -74 octal (= -60 decimal) after each clock interrupt, hence there is one clock interrupt each second.

This routine determines if any of the important times have been reached and branches to the appropriate routine. For example, at the end of a counting period, the program branches to location CLOCK2. Sub-routine CLK2 reads the scalers; ZAPPER clears scalers and ADC; PROCES processes the partially filled data buffer region; the scaler storage FWA is incremented by 24_{10} locations; and the data (spectrum) storage FWA is incremented by 1024_{10} locations.

```

CLOCK  CLOF          /CLOCK INTERRUPT COMES HERE
        ISZ 6
        JMS SETCLK
        LAC 6
        SAD THREE
CLOCKB  OPR
        SAD, TMTB
        JMP 103
CLOCKB  CLON
        JMP DISMIS
CLOCKB  ISZ OPENDR
        JMP CLOCK3
CLOCK4  LAC 7          /OPEN HUTCH 2 DOOR
        JMS DOIT
        EXC-B18-F21-BUNN /DISABLE, OR CLEAR BIT
        LAC ,CLOCK
        TCA          /RESETS TIME DUE TO OPENING OF HUTCH 2 DOOR
        DAC 7          /PRIOR TO FIRST COUNT INTERVAL
        =1
        TAD 6
        DAC 6          /BOOKKEEPING!
        LAC OPRLOC
        DAC CLOCKB
        JMP CLOCK2

```

```

CLOCK3  IS2 CCT1
        SKP
        JMP STOPIT
        IS2 THT8
        IS2 FRSTP
        JMP CLOCK2
        CLA: IAC          / LAC ONE
        JMS DCH           /START ADC
        LAC .B228
        CAM 24
        JMP CLOCK8
CLOCK2  CLON
        JMS CLK2
        JMS HAPPER
        LAC .B228
        CAM 24
        JMS PROCES
        LAC SCLOC1
        AAC 38
        DAC SCLOC1
        LAC PSPECT
        TAD .C2H8C
        DAC PSPECT
        JMP DISMIS
TSEQ    LAC 6
        JMS PRNTIN
        JMP CRLFOM
ENDIRR  CLOF
        JMS SETCLK
        DCH 6             /END OF IRRADIATION PERIOD
        CLA              /CHECK FOR SOURCE OR RABBIT
        SAD CPENDR
        JMP ENDORR1
        LAC .CLOCK       /REDUCE TIME INTERVAL TO OPEN HUTCH 2 DOOR
        TAD 7
        DAC 7
        LAH 12
        JMS DOIT         /SEND RABBIT IO HUTCH 2
        EXC+B18+F18+BUNH
ENDORR1 LAC .JCLK
        DAC ICLK
        CLON
        JMS CRLF
        LAC .BEO1
        JMS STRASC
        JMP CRLFOM
IRTIME  E
CLK2    XX
        LAC .ICAM
        CAM 24
        JMS SCALAR
        EXIT CLK2

```

Q. Processor and Data Overflow List

The first step is to set the FWA of the other data buffer region. Then the data already accumulated are checked for the presence or absence of the Tag bit. The remaining 9 bits then determine the location relative to the FWA address of the spectrum being updated. The contents of this location are incremented by one. If this is the 262144th addition, the contents of the location will be zero, and the program branches to OVRFLO. This routine sets up a table of relative channels which have overflowed and how many overflows for that channel.

```

PROCES  XX
        LAC TABLEC+2
        CAN 24
        CAN 92
        PAL
        SMO BUFFER
        JMP PROCES
        LAC BUFFER
        PAX
        LAC .BUFN1
        SMO BUFFER
        TAO .BUFS2
        DAC BUFFER
        LAC .TAGC1
        JMS CANAC2
        OGA
PROCS   LAC 0,X
        AND .DMASK
        TAO FSPECT
        DAC PROC3
        LAC 0,X
        SNA:RTL
        JMS TAG1
        ISZ PROC3
        SKP
        JMS OVRFLO
        AXS 1
        JMP PROC3
        EBA
        JMP PROCES
TAG1    XX
        LAC .SPECS
        TAO PROC3
        DAC PROC3
        JMP TAG1
PROC1   0
PROC2   0
PROC3   0
TY      0      /INITIALIZING ROUTINE
OUT9    0      /GETOTF ROUTINE
OUT06   0      / DITTO
OUT11   0

```

```

/
OVRFL0  XX          /HANDLE OVERFLOWS
        LAC ,OVDAT
        OAC OVC2
        LAC OVCTR
        SAD OVMAX
        EXIT OVRFL0
        CMA
        OAC OVC
        LAC PROC3
        TAD ,MSLOC
        ISZ OVC
        SKP
        JMP OVRFL1
        SAD OVC2
        JMP OVRFL1.2
        ISZ OVC2
        ISZ OVC2
        JMP ,.5
OVRFL1  ISZ OVCTR
        OAC OVC2
        ISZ OVC2
        ISZ OVC2
        EXIT OVRFL0
OVC     3
OVC2    0
OVTYPE  LAC OVCTR
        CMA
        OAC OVC3
        LAC ,OVDAT
        OAC OVC4
OVTY    ISZ OVC3
        SKP
        JMP CRLFDM
        LOI OVC4
        JMS PRNTIN
        ISZ OVC4
        LOI OVC4
        JMS PRNTIN
        JMS CRLF
        ISZ OVC4
        JMP OVTY
OVMAX   OVERMX

```

R. Read Data from or Write Data onto DECTAPE

All of a run may be read back into memory by typing in a large number when the CP asks for "SPECTRUM NO". If a number $\leq$ the number of spectra (time intervals) is typed, then the CP asks for "MEMORY LOCATION". A number between 1 and 17 may be typed in, which doesn't have to be the same as the SPECTRUM NO. This feature is used to facilitate overlap display of data taken for different runs.

The subroutine DTPREP shows the ordering of information dumped on the DECTAPE.

```

GETDTP  LAC .C400      /GET PORTION OF RUN FROM DECTAPE
        JMS LOCK
        JMS SETUNT     / SET UNIT TO READ FROM
        LAC OUTD3-1
        DAC DT00
        JMS CTRUN
        JMS OUTDTP
        LAC .LOGO1
        LMO
        -13
        JMS TYP70T
        LAM GETMS1-1
        JMS NMSTAS
        JMS NGET
        SKP
        JMP OUTDX      /GET ALL OF RUN
        DAC WHCHGT
        LAM GETMS2-1
        JMS NMSTAS
        JMS NGET
GETDTA  TAD .TAB1
        DAC OUTD6
        TAD .NUMSP
        DAC DT074
        LAM
        DAC DT070
        DAC DT075
        TAD DT074
        DAC OUTS
        JMS DTREAD
OUTD3   LAC UNITDT
        DATLOG-1
        @61023
        OPR
        OPR
        LAC OUTS
        JMS GET2K
        LAC DATLOG
        SAD WHCHGT
        JMP OUTD4
        ISB DT070
        JMP OUTD3-1
        ISB DT075

```

```

      JMP OUTERR
      LAC DATLOG+1
      TCA
      DAC DTD78
      ISZ DTD78
OUTERR JMP OUTD3-1
      LAM OUTEMS+1
      JMS NMSTAS
      LAC WHCHGT
      JMS PRNTSZ
      JMP OPROUT
OUTDX  LAC .JNOUTZ
      DAC OPROUT
      CLAIAC
      DAC OUTOXA
      DAC WHCHGT
      JMP GETDTA
OUTOXA 0
OUTD3  LAC OUTOXA
      SMO DATLOG+1
      JMP OUTOZ1
      ISZ OUTOXA
      ISZ WHCHGT
      LAC OUTOXA
      JMP GETDTA
OUTOZ1 LAC OPRLOC
      DAC OPROUT
      JMP OPROUT
OUTD4  LAC .EATL1
      AAC+1
      DAC 14
      LAC+ 14
      DAC GETB
      ISZ 14
      LAC WHCHGT
      AAC+1
      TAD .TIMTA
      DAC DTD11
      LAC+ 14
      DAC+ DTD11
      ISZ DTD11
      LAC+ 14
      DAC+ DTD11
      LAC+ 14
      DAC IRTIME
      LAM
      TAD+ OUTD6
      DAC 13
      -J8
      DAC SCL1
      LDI 14
      STI 13
      ISZ SCL1
      JMP .-3
      LDI 14
      DAC REJECTS
      LDI 14
      DAC OVCTR
      SNA
      JMP OPROUT
      RCL
      TCA
      DAC OVC
      LAC .OVDA1
      DAC 13
      LDI 14
      STI 13
      ISZ OVC
      JMP .-3
      JMP OPROUT

```

/GETS REJECTS

```

OUTENS .SIXBT .DIDNT FIND SPECTRUM NO 0.
GETMS1 .SIXBT .SPECTRUM NO 0.
GETMS2 .SIXBT .MEMORY LOCATION 0.
OUTOTP XX
      LAC UNITOT
      CHA
      DAC .+2
DTDO   JMS DTREAD
      -7
      IDNT-1
      OPR
      XCT DTDO
      LAC UNITOT
      LOGOUT-1
      811288
      OPR
      OPR
      EXIT OUTOTP
SET2K  XX /GET OR DUMP 1024 CHANNELS (82000 OCTAL)
      DAC OUTD2+1 / ITS DONE 256 CHANNELS AT A TIME
      LAC .8177
      DAC OUTD2+2
      -4
      DAC DT06
      XCT DTDO
OUTD2  LAC UNITOT
      7777
      177888
      OPR
      OPR
      ISZ DT06
      JMP OUTD5
      LAC .8811
      DAC OUTD2+2
OUTD5  LAC OUTD2+1
      AAC 374
      DAC OUTD2+1
      LAC DT06
      SZA
      SPA
      JMP OUTD2-1
      EXIT GET2K
DUMPER LAC .C488
      JMS LOCK
      JMS SETUNT /SET UNIT TO DUMP ON
      LAC DT07-1
      DAC DT00
      JMS OUTOTP /DUMP ALL DATA ON DEC-TAPE
      OEN WHCHDM
      LAC CCT8
      TCA
      DAC DT078
      JMS DTDUMP
      ISZ WHCHDM
      ISZ DT078
      JMP .-3
      JMS DTMRIT /NOW END OF FILE
      LAC UNITOT
DT07   8
THOUSS 1885
      OPR
OPROUT  OPR
      LAC .C488
      JMS UNLOCK
      JMP CRLFDM
OTPREP  XX
      LAC WHCHDM
      TAD .TABLE
      DAC DT074
      TAD .NUM8P

```

```

DAC DT073
LAW
YAD= DT074
DAC 13
LAC .DATL1
DAC 14
LAC WHCHON
IAC
STI 14
LAC CCT0
STI 14
LAC RUNNUM
STI 14
LAC WHCHON
YAD .TINTA
DAC DT011
LDI DT011
STI 14
ISZ DT011
LDI DT011
STI 14
LAC IRTIME
SII 14
-30
DAC SCL1
LDI 13
STI 14
ISZ SCL1
JMP .-3
LAC 6
STI 14
LAC OVCTR
STI 14
SZA
SPA
JMP= DTPREP
RCL
TCA
DAC OVC
LAC .OVDA1
DAC 13
LDI 13
STI 14
ISZ OVC
JMP .-3
OZH= 14
JMP= DTPREP
DT0UMP XX
JMS DTPREP
JMS DTHRT
LAC UNITOT
DATLOG=1
061300
OPR
LAW
YAD= DT073
JMS GET2K
EXIT DT0UMP

```

/SPECTRUM NUMBER IS FIRST ENTRY IN THIS BLOCK
 /THEN TOTAL NUMBER OF SPECTRA
 /THEN RUN NUMBER
 /THEN TWO TIME LIMITS
 /THEN IRRADIATION TIME
 /FOLLOWED BY THE SCALERS WHICH ARE
 / IN DOUBLE PRECISION
 /FOLLOWED BY NUMBER OF REJECTS
 /THEN THE NUMBER OF OVERFLOWS
 /FOLLOWED BY THE OVERFLOW LIST--CHANNEL - DATA

/DEC-TAPE DUMP OF SCALER, OTHER INFO, SPECTRA

S. Direct Dump to PDP-10

This routine parallels the DECTAPE dump. The major difference is the packing of the data before transmission.

/SEND THIS LOVELY DATA TO THE SAINT PUPLE TIMESHARING SYSTEM,
/ORR PDP15 TO ORELA PDP15 TO ORELA PEG TO PDP15
/THIS ALL MAY WORK SANS JACK HARVEY AT ORELA!!
/MAY THE GOOD LORD AND MA BELL HAVE MERCY UPON US,
/

```

SEND18 JMS BLOCK1
      LAC 10
      AND .C77
      DAC OTDUMP
      CLA
      JMS BUFBK
      LAC 10
      SAG OTDUMP
      SKP
      JMP SEND18+1
      LAC .LOGO1
      DAC 13
      -28
      JMS BUSTER
      OZM WHCMOH
      LAC CCTB
      SNA
      JMP SENDL
      TCA
      DAC DT078
      JMS ONESPE
      ISZ WHCMOH
      ISZ DT078
      JMP .+3
/ SEND LAST BLOCK
SENDL  LAM 68
      DAC 10
      CLA
      JMS BUFBK
      -388
      SAG ASCIN
      SKP
      JMP .+3
SENDEX LAM 3          /DISABLE INTERRUPTS
      JMS DOIT
      EXC+B18+F23+ASCHN+A13
      LAC .C400
      JMS UNLOCK
      LAC .LACEX
      DAC INTYPE
      JMP CALFDM
BLOCK1 XX
      LAC .BUFN1
      DAC 16
      LAM 1
      DAC+ 16
      LAM 64
      DAC+ 16
      LAC .BUFN1
      AAC+3
      JMS ASCIN
      SE
      LAM HSB10A=1
      LAM 128
      DAC 10

```

```

BLOCKA  LAC .BUFN1
        AAC -1
        DAC 16
        DZM TALLY
        -48
        DAC THP
        LAC- 16
        SAD .C168
        JMP BLOCKB
        ISZ TALLY
        ISZ THP
        JMP .-3
BLOCKB  LAC TALLY
        AAC 2
        TCA
        DAC TALLY
        LAC 16
        AAC-1
        DAC 16
        LAH 212
        DAC- 16
        LAH 215
        DAC- 16
        LAC .C388
        TAD TALLY
        TCA
        DAC ASCIN
        JMP . BLOCK1
RESPOND XX
        LAH 2          /DISABLE XMIT INTERRUPT
        JMS DDIT
        EXC-B18+F23+ASCHN+A13
        LAC .TENER
        DAC INTYPE
        JMS ASCKBR
        SAD .C28
        SKP
        JMP .-3
        JMS ASCKBR
        SAD .C177
        JMP RESPOA
        ISZ RESPOND
        SNA
        ISZ RESPOND
RESPOA  LMQ
        LAC .DISH1
        DAC ASCKBR
        LACQ
        JMP . RESPOND
ONESPE  XX
        JMS DTPREP
        LAC .DATL1
        DAC 13
        -18C
        JMS BUSTER
        -48
        JMS BUSTER
        LAC .DT075
        AAC-1
        DAC 13
        -28
        DAC DT06
        -188
        JMS BUSTER
        ISZ DT06
        JMP .-3
        JMP . ONESPE
BUSTER  XX
        DAC TALLY
BUSTA  LAC- 13
        LMQ
        CLA
        LLS+6
        JMS SUPBLK

```

```

CLA
LLS-6
JMS BUFBLK
CLA
LLS-6
JMS BUFBLK
ISE TALLY
JMP GUSTA
JMP- BUSTER
BUFBLK XX
DAC- 16
ISE ASCIN
JMP- BUFBLK
BUFBLA LAN 3
JMS D01T
EXC-010.F10-ASCHN-A13
LAN 377
JMS ASCTEL
LAC ID
JMS ASCTEL
LAC .BUFN1
DAC 16
-300
DAC ASCIN
LAC- 16
JMS ASCTEL
ISE ASCIN
JMP .-3
-3
DAC ASCIN
LAN 176
JMS ASCTEL
ISE ASCIN
JMP .-3
JMS RESPOND
JMP BUFBLA
JMP TENERR
LAC .BUFN1
DAC 16
-300
DAC ASCIN
LAC ID
AND ,C77
DAC ID
JMP- BUFBLK
ID 120
MSG10A .SIXTY /FILE.EXT[PPN]0/
MSG10B .SIXTY /ERROR CODE 0/
.TENERR SIXEX
SIXEX SLD ,C203
SKP
JMP DISMIS
TENERR DAC ID
JMS ORLF
LAC TWELVE
DAC TMP
-10
TCA
DAC TWELVE
DAC TWELV1
LAN MSG10B-1
JMS MMSTAS
LAC ID
JMS PRNTSE
LAC TMP
DAC TWELV1
DAC TWELV1
JMP BENDX

```

T. Floating-Point Routines

The DEC floating-point package⁶ for PDP-15 with EAE (extended arithmetic) was used for standard floating point operations. We used the following functions from this package:

NAME	OPERATION
ZFUNF.	Unfloat the floating accumulator. Returns with an integer. Returns with zero if the exponent is negative.
ZFNEG.	Negate the floating accumulator.
ZFNOR.	Normalize the floating accumulator.
ZFH.	Hold (Save) the floating accumulator.
ZFS.	Strip and save signs of both the floating and saved accumulators.
ZSWICH.	Switch the floating and saved accumulators.
ZFA.	General floating add.
ZFM.	General Floating multiply.
ZFD.	General floating divide.

Our contributions are ZFLT24, ZFLT., ZFLTS., ZGET., and ZSTOR. Note that in place of the DEC subroutine ZANEG, the present code uses the single-word operation TCA.

The text accompanying the routines is self-explanatory. The subroutines ZGET and ZSTOR are used for the PDP-15 FORTRAN-IV format of a floating-point number, which is an unusual format in that the least significant bits of the mantissa are in the most significant location of the two-word format.

```

/      NEW ROUTINES ADDED FOR ORCODE -- DEC 74
//      XFLT24      FLOAT POSITIVE 24-BIT INTEGER
//      XFL18      FLOAT POSITIVE 18-BIT INTEGER
//      XFLS17      FLOAT SIGNED 17-BIT COMPLEMENT 17-BIT INTEGER
//      XFLT18      GET FORTRAN-STYLE FLOATING POINT NUMBER
//      XSTOR18     STORE FORTRAN-STYLE F.P. NUMBER
//
/ITER-S:
/      XFA1        EXPONENT-FLOATING POINT ACCUMULATOR
/      XFA2        SIGN-HIGH-ORDER MANTISSA-FLT ACC
/      XFA3        LOW-ORDER MANTISSA-FLT ACC
/
XFLT24  XX          /POSITIVE 24-BIT INTEGER TO FP ARRAY
        DAC XJ2      /LOW-ORDER ADDRESS (LOA) IN AC AT ENTRY
        LOI XJ2      /HIGH-ORDER ADDRESS = LOA+1
        DAC XFA3     / TO CORRESPOND TO THE 24-BIT READ-IN
        ISZ XJ2      / FOR THE NUCLEAR DATA SYSTEM AT THE ORR
        LOI XJ2
        DAC XFA2
        LAC JC43     /24-BIT EXPONENT (VS 21 IN ORIG. CODE)
        DAC XFA1
        JMS XFNOR     /NORMALIZE THE INTEGER
        EXIT XFLT24

/-----
/
XFLT.   XX          /POSITIVE 18-BIT INTEGER TO FP ARRAY
        DAC XK1      /INTEGER IN AC AT ENTRY
        DEM XK1+1    /ZERO HIGH-ORDER BITS
        LAC XK1      /GET LOA FOR PREVIOUS ROUTINE
        JMS XFLT24
        EXIT XFLT.

/-----
/
XFLTS.  XX          /17-BIT SIGNED 2S COMPL. INTEGER
        DAC XK1+2    / TO FP ARRAY USING XFLT. ROUTINE
        SPA
        JMP J+3      /JMP IF NEGATIVE
        JMS XFLT.    /IF POSITIVE USE XFLT. DIRECTLY
        EXIT XFLTS.
        TCA
        AND .B377    /IN CASE ORIGINAL AC WAS -8 (IE 480038)
        JMS XFLT.
        LAC XNCT
        SAD XK1+2    /SET SIGN, UNLESS ORIGINAL AC WAS -8
        EXIT XFLTS. / IN WHICH CASE LEAVE AS +8
        FOR XFA2
        DAC XFA2
        EXIT XFLTS.
XK1     .BLOCK 3

/-----
/
/NEXT IS GET (STORE) FROM (TO) FORTRAN-STYLE FLOATING POINT AND
/ 3-WORD FLOATING-POINT PACKAGE ARRAY
/ FORTRAN-STYLE FLOATING POINT IS 2-WORD ARRAY AS FOLLOWS:
/ 1ST WORD  2  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
/ [LOWEST ORDER BITS OF MANTISSA] * [2S COMPL. EXPONENT]
/
/              SIGN OF EXPONENT
/
/ 2D WORD  8  1  2...17
/ * [HIGHEST ORDER 17 BITS OF MANTISSA. AC1 ALWAYS 1
/   IN PROPERLY NORMALIZED REPRESENTATION]
/ * SIGN OF MANTISSA.
/ MANTISSA IS SIGN-MAGNITUDE, NOT 2S COMPL. NOTATION
/-----

```

```

/
XGET.  XX          /ENTER WITH ADDRESS OF FIRST WORD OF FORTRAN.
      DAC XJ3      / STYLE F.P. NUMBER
      LDI XJ3
      AND .B7772
      DAC XFAC3     /GET 9 LOW-ORDER BITS
      LDI XJ3
      SWHA
      SPA:CLL
      STL
      LRS 1:
      DAC XFAC1     /GET EXPONENT IN 2S COMPLEMENT
      ISZ XJ3
      LDI XJ3
      DAC XFAC2     /GET 18 HIGH-ORDER BITS, INCLUDING SIGN BIT
      EXIT XGET.
/
/-----
/

```

```

XSTOR. XX          /ENTER WITH ADDRESS OF FIRST WORD OF FORTRAN.
      DAC XJ3      / STYLE F.P. NUMBER
      LAC XFAC1     /GET EXPONENT
      SPA
      TAD .B7774    /TEST FOR EXPONENT OVERFLOW
      SPA
      JMP XPOK
      CL: IAC       /IF EXPONENT OVERFLOW SET FLAG=1
      DAC XGVFLG    / AND DATA=0
      DEM XJ3
      ISZ XJ3
      DEM XJ3
      EXIT XSTOR.
XPOK.  LAC XFAC1     /EXPONENT OK, GET INFO TO STORE
      AND .C777
      STI XJ3       /STORE EXPONENT
      LAC XFAC3
      AND .B777
      XOR XJ3
      STI XJ3       /STORE 9 LOW-ORDER BITS WITH EXPONENT
      ISZ XJ3
      LAC XFAC2
      STI XJ3       /STORE HIGH ORDER BITS INCLUDING SIGN
      DEM XGVFLG
      EXIT XSTOR.
XGVFLG ?
/
/-----
/

```

```

/
/      TEMPORARY STORAGE
/
XFAC1  0
XFAC2  0
XFAC3  0
XHAC1  0
XHAC2  0
XHAC3  0
XJ1    0
XJ2    0
XJ3    0
XJ4    0
XJ5    0
XSIGN  0          /SIGN OF FLOATING ACC
/
/-----
/

```

U. Nuclear Data 5050 Analyzer Dump

Data may be accumulated in the analyzer at the same time data are being accumulated and stored in the computer. The data stored in the analyzer are usually the results of Ge(Li) spectroscopy which are subsequently analyzed using a program written in FORTRAN-IV and compiled by the PDP-15 compiler. The following routine is used to prepare a DECTAPE dump in the proper format to be read by the analysis code. Included in this preparation is a determination of ADC deadtime. Another routine allows decay time to be input in hours, days..., and converts this time to seconds.

```

.NOLST
/ NEXT IS TO DUMP JUEL EMERY DATA FROM NUCLEAR DATA ANALYSER
/ IN FORMAT SO THAT HE CAN LATER READ IN WITH HIS MONSTER PROGRAM
/
EMERY   LAC .DATLO
        JMS ASCIN
        11
        LAM JFEMS1=1
        =3
        JMS LOGFIX
        LAC .DATL1      /SET UP TO WRITE FILENAME IN SIXBIT
        JMS IDNTPC      /DOES FIRST 2 WORDS
        ISZ LY1
        DCM LY1
        JMS ID3
        JMS ID3
        JMS ID3          /FILENAME IN IDNT 3-WORD ARRAY
        LAC .HXCT
        DAC IFL          /FIRST WORD OF FIRST DUMP
        =4
        DAC TALLY
        LAC .IFL
        DAC 14
        LAC .LOGOU
        DAC LY1
        LDI LY1
        STI 14
        ISZ LY1
        ISZ TALLY
        JMP =4
        DCM IFL+5        /FILENAME IN 1-BIT FOR FIRST DUMP
        DCM IFL+6
        DCM IFL+7
        DCM IFL+10       /THESE LAST TO BE TIDY
        JMS ORLF
        LAC .DATLO
        JMS ASCIN
        67
        LAM JFEMS2=1    /ASCII IDENTIFICATION INPUT
        =13             / LIMITED TO 25 CHARACTERS (27 OCTAL)
        JMS LOGFIX
        =26
        DAC TALLY
        LAC .IFL10
        DAC 14
        LAC .LOGOU

```

```

DAC LY1
LDI LY1
STI 14
ISE LY1
ISE TALLY
JMP .-4
JMS CRLF
JMS TYPEIN
LAC .IFL37
JMS RSTOR.
JMS CRLF
LAW JFENS3-1
JMS NHSTAS
JMS TYPEIN
AAC-1
DAC IFL+53
JMS CRLF
JMS TYPEIN
DAC IFL+94
JMS CRLF
LAW JFENS4-1
JMS NHSTAS
JMS TYPEIN
DAC IFL+6-
JMS CRLF
JMS TYPEIN
DAC IFL+61
JMS CRLF
/NEXT TAKEN FROM JFE CODES
CLA
DAC BEG
-1888E
DAC CCTR
JMS RANGE
LAC BITS
SAD TWO
JMP .-4
LAC .JMPWR
DAC WRD
JMP .-3
LAC OPRLOC
DAC WRD
LAC .SLOW1
DAC 14
LAC BEG
ONBSF
JMP .-1
CSARR
SKP
WRTF2 RRRR
ISE BEG
RQMR
STI 14
WRD NOP
ROMH
JMS ROTACR
STI 14
WRD1 ISE CCTR
JMP WRTF2
COFF
/NEXT GET SUM OF DATA, WEIGHTED SUM OF DATA STARTING FROM
/ CHANNEL 28 (SAY), FIRST CHANNEL HAS INTEGER LIVE TIME.
LDI .SLOC1
IAC
JMS RPLT.
LAC .IFL43
JMS RSTOR.
-1888E+28
DAC TALLY
LAC .C28
CLL
MUL
BITS 1

```

/GET "GEOMETRY", A FLOATING POINT NUMBER
 /GET NGR...SUI "WHERE DOES IT GO??
 /MAYBE HERE?
 /GET MAX GRP
 /GOES HERE FOR SURE
 /GET "LOW" CHANNEL
 /GET "HIGH" CHANNEL
 /ASSUME 4096 CHANNEL SPECTRUM
 /- NO OF CHANNELS
 /BITS=1 FOR 18-BIT (MAX) DATA
 / =2 FOR 24-BIT DATA
 /FOR 24-BIT DATA
 /COMPUTER ON (TO ANALYZER)
 /WE ARE INTO THE "WRIT" OF JFE CODE
 /OR JMP WRD1 FOR 18-BITS
 /DOUBLE-PRECISION INTEGERS ARE STORED
 / LOW-ORDER FIRST, THEN 6 HIGH ORDER
 /DATA ALL IN MEMORY, COMPUTER OFF
 /LIVE TIME ASSUMED 18 BITS
 / JFE ADDS 1
 /LIVE TIME IN FLOATING POINT
 /PUT INTO IFL-ARRAY
 /SUMMIN COUNTER


```

JFLOOP LACQ
DAC PROC1      /-28 FOR 18 BITS, 48 FOR 24-BITS
TAD .SLOC1
DAC PROC2      /FIRST LOCATION OF DATA
OZM DATSUM
OZM DATSUM+1
OZM WTSUM
OZM WTSUM+1
LAC BITS
SAD TWO
JMP .+4
LOI PROC2
JMS XFLT.
JMP .+4
LAC PROC2
JMS XFLT24
ISZ PROC2
JMS XFH.
LAC .THOLD
JMS XSTOR.
LAC .DATSU
JMS XGET.
JMS XFA.
32
LAC .DATSU
JMS XSTOR.
LAC .THOLD
JMS XGET.
JMS XFH.
LAC PROC1
JMS XFLT.
JMS XFH.
JMS XFH.
LAC .WTSUM
JMS XGET.
JMS XFA.
32
LAC .WTSUM
JMS XSTOR.
ISZ PROC1
ISZ PROC2
ISZ TALLY
JMP JFLOOP
JMS XFH.
LAC .ADCON
JMS XGET.
JMS XFH.
LAC .WTSUM
JMS XSTOR.
LAC .DATSU
JMS XGET.
JMS XFH.
LAC .IPL43
JMS XGET.
JMS XFD.
JMS XFH.
LAC .CTRAT
JMS XSTOR.
LAC .TALJP
JMS XGET.
JMS XFH.
JMS XFNEG.
JMS XFH.
CLA: IAC
JMS XFLT.
JMS XFA.
32
LAC .SHTCO
JMS XSTOR.
LAC .DATSU
JMS XGET.
JMS XFH.

```

/FOR 18-BIT DATA
 /GET ADDRESS FOR 24 BIT DATA
 /24-BIT DATA IN 2 WORDS
 /DATUM IN HOLY-ARRAY
 /SAVE VALUE
 /GET CURRENT SPECTRUM SUM
 /ADD NEW DATUM
 /AND PUT BACK INTO SPECTRUM SUM
 /RETRIEVE DATUM
 /GET CHANNEL NUMBER
 /MULTIPLY-- HAVE (CH. NO.)*(DATUM)
 /GET CURRENT WEIGHTED SUM
 /ADD TO WTSUM
 /AND PUT BACK INTO WTSUM
 /WTSUM TO HOLY
 /GET ADC CONSTANT
 /MULTIPLY
 /AND STORE IN WTSUM
 /GET TOTAL NUMBER OF COUNTS
 /DIVIDE BY LIVE TIME
 /DIVIDE BY LIVE TIME
 /AND STORE AS COUNT RATE
 /GET CONSTANT KNOWN AS "TAL" IN JPE CODE
 /TAL*CTRATE
 /SAME AS LAC {1
 /NOW HAVE (1.*TAL*CTRATE)
 /STORE IN SHT-CORRECTION
 /GET TOTAL NO OF COUNTS

```

LAC .CFOT      /DEAD TIME CORRECTION
JMS XGET.
JMS XFN.       /"CORRECTED" TOTAL COUNTS
JMS XFN.
LAC .WTSUM     /GET WEIGHTED SUM
JMS XGET.      /20 PART OF TIME CORRECTION
JMS XFA.
32
JMS XFN.
LAC .IFL43     /SET LIVE TIME
JMS XGET.
JMS XFA.       /ADD CORRECTION
32
LAC .IFL45     /"CLOCK" TIME
JMS XSTOR.
LAC JFENS7-1
JMS NHSTAS
JMS PPTYP
LAC JFENS8-1
JMS NHSTAS
LAC .IFL43     / NEXT IS LIVE TIME
JMS XGET.      /TYPE IN FIXED POINT
JMS XFUNF.
JMS PRNTIN
LAC .CTRAT
JMS XGET.
JMS PPTYP
LAC JFENS8-1
JMS NHSTAS
JMS CRLF
LAC JFENS5-1   /NEXT MESSAGE BEFORE TYPE IN
JMS NHSTAS
JMS TYPEIN
SEA
JMP .+3
IAC
JMS XFLT.
JMS XFN.       /IF SMT TYPEIN WAS ZERO, SET #1.
LAC .SMTCO     /GET CORRECTION
JMS XGET.
JMS XFN.       /SMT CORRECTED
LAC .IFL41
JMS XSTOR.     /PUT INTO IFL ARRAY
JMS CRLF
JMS TYPEIN     /VALUE OF DECAY TIME IN SOME UNITS
JMS XFN.       /HOLD IT TO GET UNITS
JMS CRLF
INTAG JMS INTYPE /TO CORRESPOND TO JPE, TYPE S,M,H,D,Y
SAD .C323      /32308(SECONDS)
JMP SECEND
SAD .C319      /31908(MINUTES)
JMP MINEND
SAD .C318      /31808(HOURS)
JMP HREND
SAD .C304      /30408(DAYS)
JMP DAYEND
SAD .C301      /33108(YEARS); TOSSED IN ALTHOUGH I DOUBT
JMP YREND     / THAT IT WOULD BE USED
LAC 277       /IF IT GETS TO HERE ITS A GOOF
JMS TYPOT
JMS CRLF
JMP INTAG
YREND LAC .NDAY
JMS XGET.
JMS XFN.
JMS XFN.       /MULTIPLY BY YEARS
LAC .C30      /24 HOURS/DAY
JMS XFLT.
JMS XFN.
JMS XFN.

```

```

WREND   LAC .C74           /60 MINUTES/HOUR
        JMS XFLT.
        JMS XFM.
        JMS XFM.
MINEND   LAC .C74           /60 SECONDS/MINUTE
        JMS XFLT.
        JMS XFM.
SECEND   LAC .IFL47         /STORE IN IFL-ARRAY
        JMS XSTOR.
        JMS CRLF
/FINISH  UP IFL-ARRAY
        D2M IFL+51         /THESE D2M TO ZERO NUMBERS EQUIVALENT
        D2M IFL+52         / TO THOSE IN EXAMPLE OF JFE OUTPUT
        D2M IFL+62
        D2M IFL+66
        LAC .BTHOU
        DAC IFL+55
        AAC-1
        DAC IFL+56         /AGAIN TO CORRESPOND TO EXAMPLE
        CLA:IAC
        DAC IFL+57         /THESE IFL HAVE 1 IN THEM
        DAC IFL+64
        IAC
        DAC IFL+65         /HAS A 2 IN IT
        LAC BITS           /LAST LOCATION IN IFL-ARRAY HAS 1 OR 2
        DAC IFL+67         / ACCORDING TO 18 OR 24 BIT DATA
        LAC DISPLC+1       /LAC (12
        DAC IFL+63         /MAY BE UNIMPORTANT, BUT...
        JMS SETUNT        /GET DEC TAPE UNIT NO. TO DUMP ON
        LAC UNITOT
        CMA
        DAC .+2
        JMS DTHWRT        /INITIALIZE TAPE. CHECK FOR FILENAME
        -7                / DUPLICATION.
        IDNT=1
        OPR
        JMS DTHWRT        /DUMP THE IFL BLOCK
        LAC UNITOT
        IFL-1
        035800            / HOPEFULLY (70/2+1)*1000 IN OCTAL
        OPR
        OPR
        JMS DUMP4K        /NOW DUMP THE 4K SPECTRUM
        JMS CRLF
        LAC JFMS6
        JMS STRASC
        JMP OTD7-1        /END OF FILE
DUMP4K   XX                /DUMP 4096 SPECTRUM AS CLOSE TO JFE FORMAT
        LAC BITS          / AS WE CAN
        TCA
        TAG .SLOM2
        DAC P4K           /FIRST DUMP LOCATION -1
        IAC
        DAC OTDUMP        / (TEMPORARY STORAGE)
        IAC
        DAC GET2K         / (ANOTHER TEMP STORAGE)
        -1
        STI GET2K        /MEMORY IS OFF ONE CHANNEL IN HIS DUMP
        ISZ GET2K        / HE HAS 777777 IN THIS WORD FOR 18-BIT DATA
        LAC BITS         / PRESUMABLY SOMETHING IN THE NEXT WORD FOR
        SAD TWO          / FOR 24-BIT DATA. WE PUT IN A 2,
        STI GET2K
        SAD TWO
        AAC-21           /17 DECIMAL =61OCTAL DUMPS OF 256 DATA FOR
        AAC-22           / 18-BIT DATA--PRESUMABLY 41 OCTAL
        DAC OTD6         / DUMPS OF 256 DATA FOR 24-BIT DATA SPECTRUM
        LOI OTDUMP       /REPLACE FIRST WORD WITH HEAD2R
        DAC LRS7         / (TEMP STORE)
        LAC .MXCY        /HEADER WORD
        STI OTDUMP
        JMS DTHWRT
        LAC UNITOT

```

```

P4K      SLOC1=3
.B177    177800      / =(374/2 +1)*1000
          QPR
          QPR
          ISZ DT06
          SKP
          EXIT DUMP4K
          LAC LRS7      /REPLACE DATUM
          STI DT0UMP
          LAC P4K
          AAC=372      /MOVE 250 CHANNELS
          DAC P4K
          IAC
          DAC DT0UMP
          JMP MP4K

/
JFEMS1   .SIXBT "FILENAME"
JFEMS2   .SIXBT "TITLE, GEOM"<0>
JFEMS3   .SIXBT "GRP, MAX GRP"<37><0>
JFEMS4   .SIXBT " L AND H CH-4"<37><0>
JFEMS5   .SIXBT "S.MT, DTD, UNITS"<37><0>
JFEMS6   .SIXBT "...
JFEMS7   .SIXBT ". DTD=0,
JFEMS8   .SIXBT ". DTL=0,
JFEMS9   .SIXBT " CTS/SEC"<0>
ADCONS   734746 /ADC CONSTANT = 1.0E-5
          253630
TALJFE    277700 /AMPLIFIER RESOLVING TIME = 9.0E-6
          241037
CFBT      703761 /CONSTANT FRACTION OF DEAD TIME=10.9E-6
          236426
YRDAY     341011 / 365.24 DAYS/YEAR
          206475

/
RANGE     XX      /TAKEN FROM JFE CODE
          LAC ,ROMM
          DAC MORE+1
          LAC ,C14
          DAC ML
          D2M BITCTR
          D2M HW
RPT        -10000
          DAC NCH
          LAC BEC
          ONBSP
          JMP .-1
          CSAR
MORE        RRHR
          ROMM
          TCA
          TAD HW
          SPA
          JMP NEUMAX
INC         ISZ NCH
/           NOTE THIS LOOP COULD BE CONFUSED BY ALL ZERO DATA
          JMP MORE
          LAC HW
          SZA
          JMP FINDBT
          D2M ML
          LAC ,ROML
          DAC MORE+1
          JMP RPT
NEUMAX     XCT MORE+1
          DAC HW
          JMP INC

```

```

FINDBT  COFF
        RCL
        SZL
        JMP 03
        ISZ BITCTR
        JMP 04
        LAC BITCTR
        TCA
        TAD 022
        TAD ML
        AAC-22
        SNA:SPA
        SKP:GLA
        CLA:IAC
        IAC
        DAC BITS
        EXIT RANGE
ROTACR  XX
        LRS-6
        AND 097
        EXIT ROTACR

```

/10 BITS-ZERO BITS=WORD BITS
 / 1/2 WORD=12, AC=BITS/CHAN

/BITS BETTER HAVE A 1 OR 2 ONLY AT THIS POINT
 /HANDLES UPPER-MOST 6 BITS

V. Definition and Storage of Constants

We defined every constant used, and set up every storage location, rather than allow definitions to be determined by the assembler, for two reasons. The first was ease in mnemonics, and the second was ease in debugging. In addition, as may have been noticed by the observant reader, several storage locations were used by more than one routine, a task made easier by specifically defining locations.

```

/ST
/NEXT IS IN END4.SRC FILE
/
/NEXT ARE SHALL CONSTANTS
ONE      1
TWO      2
THREE    3
.C13     13
.C14     14      /NUMBERS 4,5,6,7,10,11, AND 12 IN TABLES
.C17     17
.C20     20
.C21     21
.C22     22
.C30     30      /32 IN JFLOOP IF NEEDED
.C37     37      /40 IN DYNRIO
.C43     43
.C74     74      /67 IN EMERY
.C77     77
.C177    177
.C203    203
.C212    212
.C215    215
.C220    220
.C222    222
.C225    225
.C253    253
.C254    254
.C255    255
.C256    256
.C260    260
.C304    304
.C305    305
.C310    310
.C315    315
.C322    322
.C323    323
.C331    331
.C377    377
.C511    511
.C1020   1020
.C1777   1777
.C2000   2000
.C7777   7777
/
/

```

/NEXT ARE BIGGER NUMBERS

```
.M2000 -2000
.MCC 468358
.ICAM 620000
.BTHOU 12000
.BMP 167240
.BMP 157240
.M777 77777
.MXCT 400000
.B204 22400
.B77 77000
.BNE 167540
.BAE 400175
.B220 220000
.BEQ1 051711
.B1000 120000
.B377 377777
.B7770 777000
/.B7774 777400 /IN ROUTINE DT00
.B011 11000
.CAE 400375
.CLCAM 720000
/30000 IN EMERY
/61000 IN OTDUMP
/177000 =LOC .B177 IN DUMP4K
//
```

/NEXT ARE OTHER CONSTANTS, E.G. LOCATIONS

```
.JMPIN JMP INTRUP
.LACEX LAC EXECIN
.INBIT 364811
.TAB1 TABLE=1
.NUM1 NUMSPC=1
.TABNU TABLE=NUMSPC
.ABCNS ASCN=SCAN
.IDNT IDNT
.SCLR SCLR
.SLOC1 SLOC1
.CENT CENT
.NBLOC =SLOC1
.SLOC1 SLOC1=1
.SLOC2 SLOC1=2
.LLS 640000
.LRS 640500
.SPECS SPECS2
.DISBU DISBU
.DMASK DMASK
.OVERH =OVERMX=2
.OVDA1 OVDATA=1
.BUFN1 BUFN1
.BUFS2 BUFS2
.EXEC3 EXECI3
.MEMMA MEMMAX
.ALS2 ALS=2
.ISSEX ISB EXPCT
.INO10 INOTTS=10
.INOT1 INOTTS
.INOT6 INOTTS=6
.EXC EXC
.NUMSP NUMSPC
.OTDA1 OTDATA=1
.OTDIR OTDIR
.OTDAT OTDATA
.OTD11 OTDIRH=1
.OT40 OTDATA=40.1
.DTCBL DTCBLK
.OTUTS OTUTST
.DATLO DATLOG
.NUMS1 NUMSPC=1
.TINTA TINTAS
.SCLR1 SCLR=1
.DATL1 DATLOG=1
.LOGOU LOGOUT
.TABCL TABLE=1
.OVDAT OVDATA
```

```

.LOGO1 LOGOUT-1
.DATL5 DATLOG-5
.TABLE TABLE
.TABM1 TABLEC-1
.TAB11 TABLEC-11
.IFL IFL
.IFL45 IFL-45
.IFL18 IFL-18
.IFL37 IFL-37
.IFL43 IFL-43
.IFL41 IFL-41
.IFL47 IFL-47
.JMPWR JMP WRD1
.THOLD THOLD
.DATSU DATSUM
.WTSUM WTSUM
.ADCON ADCONS
.CTRAT CTRATE
.TALJF TALJFE
.SWTCO SWTCOR
.CFDT CFDT
.YRQAY YRQAY
.RQMM RQMM
.RQML RQML
.CLOCK 2 /DOOR OPEN INTERVAL PRIOR TO FIRST COUNT
.K1 XK1
.JMOUT2 JMP OUTD2
.JCLK JMP CLOCK
.JENDIR JMP ENDIRR
.JINITA JMP INITA
.BCLK JMP CLOCKS
/
/NEXT ARE CONSTANTS RELATING TO CAMAC
.CH5 EXC-B18-F1-DSFN-A1 /LOG ARESET
.CH7 EXC-B18-F1-DCNN-A /CAMAC-7
.CH18 EXC-B18-F1-BUNN /BUNSI-A-1
.OTD6 6
.STRAT1 XX
.STRAT2 XX
.INOUT4 XX
.PRNTT2 XX
.N-NDN XX
.FPT1 2
.ACCN 0
.EXPCT 0
.SIGN 0
.ESIGN 0
.CO9 0
.INTMSK 0
.REG1 0
.REG2 0
.REG3 0
.REG4 0
.BF1 0
.CO0X 0
.YFLSC 0
.RSELEC 0
.UT1 0
.UT2 0
.UT3 0

```



```

BUT4      0
BUT5      0
BUT6      0
BTALLY    0
BF3       0
BF4       0
DSFLAG    0
BF4       0
SPECX     0
XLN       0
YLN       0
EXECT2    0
EXECT1    0
INTRAC    0
INTRMQ    0
CQ1       0
BTDETS    0
BTDE1     0
OPENDR    0
TINTAB    .ZBLOC NUMSPC+1
SCLR      .ZBLOC NUMSPC+30
TABLE     .ZBLOC NUMSPC+2
OVDATA    .ZBLOC OVERMX+2
DAYLOG    .ZBLOC DAYLN
LOGOUT    .ZBLOC 30
BUFN1     .ZBLOC BUF32+2
OTDATA=BUFN1
DTDIRM=OTDATA+400
IFL=DTDIRM+40
WHCHGT=IFL+70
WTSUM=WHCHGT+1
THOLD=WTSUM+2
CTRATE=THOLD+2
SWTCOR=CTRATE+2
ML=SWTCOR+2
BITCTR=ML+1
HW=BITCTR+1
NCH=HW+1
SEG=NCH+1
CCTR=SEG+1
CTD70=CCTR+1
DTD74=CTD70+1
DTD75=DTD74+1
IG1       0
IG2       0
IG3       0
IG4       0
IG5       0
IG6       0
CCT0      0
CCT1      0
FRSTP     0
REJECTS   0
DATSUM    0
          0
          .SIZE
          .END

```

ACKNOWLEDGMENTS

The grandfather of some of the present code was written about a decade ago for a PDP-7 system by Alex Marusak. Basic ideas for teletype input and output which were developed for that code were used in the present code, and we are pleased to acknowledge this source. Several of the arithmetic routines in the present code exhibit the special genius of J. A. Biggerstaff, whose contributions are always welcome. We also appreciate the work of J. G. Craven for providing software at the receiving end necessary for the direct-line data dump to the PDP-10. The coding of the routine to process data from the Nuclear Data 5050 Analyzer was made much easier because J. F. Emery graciously made his codes available. Some coding concepts of W. E. Kinney and F. G. Perey are utilized for which we express our appreciation. T. A. Love must be credited with providing the necessary user's viewpoint and for discovering many of the errors in prior versions.

APPENDIX. CAMAC INSTRUCTIONS AND PROGRAMMING EXAMPLES

In this appendix is listed all of the CAMAC programming information used in the ORCODE.77 programming. The physical slots in the CAMAC crate which contain modules used in the present experiment are as follows:

Three scaler units (of 4 scalers each) in positions 1, 2, and 3;
the direct-dump module in position 10; the display module in
position 13; the ADC module in position 14; the pneumatic-tube
control module in position 17; and the data channel control in
position 23.

CAMAC DOCUMENTS

TID-25375 CAMAC A MODULAR INSTRUMENT SYSTEM FOR DATA HANDLING
TID-25376 CAMAC ORGANIZATION OF MULTI-CRATE SYSTEM
TID-25377 SUPPLEMENTARY INFORMATION ON CAMAC INSTRUMENT SYSTEM

PDP-15 CAMAC CRATE CONTROLLER.

REGISTER ASSIGNMENTS:

0	EFNA FOR AC TRANSFERS		
1	UNUSED		
2	LAM STATUS WORD		
3	DATA REGISTER FOR AC TRANSFERS		
4	FNA		DCM A
5	CURRENT ADDRESS (CA)		DCM A
6	WORD COUNT (WC)		DCM A
7	DATA REGISTER		DCM A
10	FNA	DCM B	
11	CA	DCM B	
12	WC	DCM B	
13	DATA	DCM B	
14	UNUSED		
15	UNUSED		
16	UNUSED		
17	UNUSED		

***** IOT 503x
5001 SKIP IF CONTROLLER IS NOT BUSY AND UPDATE LAM STATUS (REG. 2)
5012 READ REG. 0 (I.E. EFNA)
5004 LOAD REG. 0

EFNA WORD FORMAT:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
'	'	'	'	<	F	>	<		N		>	<		A		>	
' ' ' ' ADDRESS SCAN																	
' ' 16 BIT MODE																	
' UNUSED ON OUTPUT																	
EXECUTE DATAPATH OPERATION																	

BITS 0 AND 1 ARE 0 AND X RESPECTIVELY ON INPUT.

..... 12T 506X

5064 LOAD REG. 3 FROM AC
5072 READ REG. 3

..... 18T 502X

5021 SKIP IF INTERRUPT ACTIVE
5032 READ 18 BITS OF THE LAN STATUS

AC0 IS LAN FROM MODULE 23, NORMALLY DCH CONTROL.

AC1 - AC17 ARE LANS FROM MODULES 17 - 1 RESPECTIVELY

5024 LOAD REG. 2
FORMAT:

```

0 1 2 3 4          14 15 16 17
* * * * *          REG. ADDRESS FOR 18T 504X
* * * * * EXECUTE FUNCTIONS IN BITS 0 -3
* * * GENERATE 2
* * * GENERATE C
* * * ENABLE INTERRUPT
INHIBIT

```

NOTE: ONE MUST BE SURE NO DATAWAY OPERATION IS IN PROGRESS
WHEN THE COMMAND TO GENERATE 2 OR C IS GIVEN.
IN PRACTICE THIS MEANS THAT THE DCH'S MUST BE STOPPED
EITHER BY WAITING FOR WORD COUNT OVERFLOW OR BY
COMMAND TO DISABLE BOTH DCH'S.

..... 18T 504X

5044 LOAD REG. REFERENCED IN PREVIOUS 18T 5024
5052 READ REG.

COMPUTER I/O RESET OR CAF INSTRUCTION SETS INHIBIT

.....
N - PHYSICAL SLOT IN WHICH MODULE IS LOCATED
F - FUNCTION CODE
A - SUB ADDRESS
M - PDP15 TO/FROM CAMAC CONTROLLER DATA TRANSFER MODE
CAM=13T 500G

DISPLAY CONTROL

```

F(16) A(0)      LOAD Y, 10 LSH
F(16) A(2)      LOAD X, 9 LSH
F(16) A(1)      LOAD X AND INTENSIFY
F(16) A(3)      LOAD Y AND INTENSIFY
F(16) A(4)      LOAD X AND Y AND INTENSIFY
                  Y = 9 MSR, X = 9 LSH
F(8) A(0)       TEST MARKER FLAG, J=1 IF FLAG "1"
F(17) A(0)      LOAD CONTROL REGISTERS

```

9 10 11 12 13 14 15 16 17

```

* * < X INCREMENT          >
* * CONNECT TO DCH 5
* * ENABLE MARKER INTERRUPT

```

F(1) A(1) READ FUNCTION PANEL

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
? ? ? ? < ? > < ? > <      ?      ><      ?      >
? ? ? ?      ?      ?      ?      ?      Y FULL SCALE
? ? ? ?      ?      ?      ?      REGION SELECT
? ? ? ?      ?      LATCHING PUSHBUTTONS
? ? ? ?      MOMENTARY PUSHBUTTONS
? ? ? ? LOW LINE
? ? HIGH LINE
? ? MARKER DIRECTION
? MARKER FLAG
```

2.52 1) RESETS MARKER FLAG
2) DISABLES MARKER INTERRUPT
3) DISCONNECTS DCH

C.52 DOES NOTHING

BLOCK TRANSFER

F(0) A(0) READ CONTROL AND STATUS REGISTER

```

0 1          16 17
* *          * * DCH A ENABLED
* *          * * DCH B ENABLED
* * DCH B WORD COUNT OVERFLOW
* * DCH A WORD COUNT OVERFLOW

```

F(18) A(0) SELECTIVELY SET DCH ENABLES.
BIT 16 ENABLES DCH B AND BIT 17 ENABLES DCH A

F(21) A(0) SELECTIVELY DISABLE.
BIT 16 DISABLES DCH R AND BIT 17 DISABLES DCH A

F(8) A(0) TEST FOR MC OVERFLOW. Q=1 IF MC OVERFLOW HAS OCCURED ON AN ENABLED DCM

2.82 OR COMPUTER I/O RESET DISABLES ALL DCH'S AND
 CLEARS ANY PENDING DCH OPERATIONS

C.S2 DOES NOTHING

ADC INTERFACE

F(17) A(1) LOAD CONTROL REGISTER

13 14 16 17

* * * * CONNECT TO DCH A
 * * * * CONNECT TO DCH B
 * * REJECT MODE
 * * ENABLE ANALYZE

F(1) A(1) READ CONTROL REGISTER
 F(2) A(0) READ DATA REGISTER

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
 * * * * < ADC DATA >
 * * * * ALTER BIT
 * * TAG 1
 * * TAG 2

F(6) A(0) TEST BUFFER READY. 0=1, BUFFER READY
 F(27) A(0) TEST ANALYZE ENABLE. 0=1, ENABLED
 F(27) A(0) TEST DCH ENABLED. 0=1, NOT ENABLED

Z.S2 1) RESETS ADC
 2) DISABLES ANALYZE
 3) DISABLES REJECT MODE

(Z+C).S2 RESETS ADC

INHIBIT (I) STOP S DATA TRANSFER TO COMPUTER BUT DOES NOT
 RESET OR DISABLE THE ADC.

JORNAY MODEL 85A SCALER

F(0) A(0) READ SCALER 1
 F(0) A(1) READ SCALER 2
 F(0) A(2) READ SCALER 3
 F(0) A(3) READ SCALER 4

F(9) A(0) CLEAR SCALER 1
 F(9) A(1) CLEAR SCALER 2
 F(9) A(2) CLEAR SCALER 3
 F(9) A(3) CLEAR SCALER 4

F(25) (A(0)+A(1)+A(2)+A(3)) INCREMENTS ALL FOUR SCALERS FOR TEST

Z.S2 CLEARS ALL SCALERS

C.S2 CLEARS ALL SCALERS

INHIBIT (I) CONTROLS GATING OF ALL CHANNELS WHEN FRONT PANEL
 GATING SWITCH IS IN APPROPRIATE POSITION

HARE CONTROLLER

F(1) A(0) READ COMMAND AND STATUS REGISTER AND CLEAR INTERRUPTS
 F(19) A(0) SELECTIVE SET COMMAND REGISTER
 F(21) A(0) SELECTIVE CLEAR COMMAND REGISTER
 F(8) A(0) TEST FOR INTERRUPT 0=1 TRUE
 F(27) A(1) TEST FOR BUNNY IN HUTCH #2 0=1 TRUE
 F(27) A(2) TEST FOR BUNNY IN REACTOR 0=1 TRUE

 Z.S2 CLEARS COMMAND REGISTER AND INTERRUPTS
 C.S2 DOES NOTHING

COMMAND AND STATUS REGISTER FORMAT

9 10 11 12 13 14 15 16 17

* * * * * SEND BUNNY TO REACTOR
 * * * * * RETURN BUNNY TO HUTCH
 * * * * * CLOSE HUTCH #2 DOOR
 * * * * * REMOTE CONTRL
 * * * * * HUTCH #1 DOOR OPEN
 * * * * * BUNNY IN HUTCH #2
 * * * * * ENABLE BUNNY IN HUTCH #2 INTERRUPT
 * * * * * BUNNY IN REACTOR
 * * * * * ENABLE BUNNY IN REACTOR INTERRUPT

ASYNCHRONOUS COMMUNICATIONS MODULE

F(0) A(0) READ RECEIVER REGISTER
 0 1 10 11 12 13 14 15 16 17
 * * * * * DATA *
 * * * * * PARITY *
 * * * * * BREAK *
 * * * * * PARITY ERROR *

 F(16) A(0) LOAD TRANSMIT REGISTER

 F(19) A(13) SELECTIVELY ENABLE INTERRUPTS
 F(23) A(13) SELECTIVELY DISABLE INTERRUPTS

 FORMAT:
 16 17
 * *
 * * RECEIVER DATA READY
 * * TRANSMITTER BUFFER READY

 F(8) A(15) TEST ENABLED INTERRUPTS, 0=0 NO INTERRUPT
 0=1 SOME INTERRUPT

 F(1) A(12) READ STATUS REGISTER
 F(1) A(14) READ INTERRUPT REGISTER
 FORMAT:
 16 17
 * *
 * * RECEIVER DATA READY
 * * TRANSMITTER BUFFER READY

 BREAK MEANS NO STOP BIT WAS RECEIVED.
 OCCURS IF LINE IS OPEN

 Z.S2 CLEARS RECEIVE DATA READY, SETS TRANSMIT BUFFER READY,
 AND DISABLES ALL INTERRUPTS.
 C.S2 DOES NOTHING

PROGRAMMING EXAMPLES:

/ACCUMULATOR INPUT/OUTPUT

/READ N=1,F=0,A=0,M=100

LAC (500020

CAM 5

JMP .-1

CAM 72

/N=1,F=0,A=0,M=100, EXECUTE

/LOAD REG. 0 AND EXECUTE COMMAND

/TO MODULE

/READ 10 LSB INTO AC

/READ N=1,F=0,A=0,M=240

LAC (400020

CAM 5

JMP .-1

CAM 72

DAC DATA

CAM 72

DAC DATA+1

/INPUT 10 LSB

/INPUT 5 MSB

/WRITE N=13,F=16,A=0,M=100

LAC (100000

CAM 5

JMP .-1

LAC DATA

CAM 64

/SET MODE TO 10 BIT

/FIRST THREE INSTRUCTIONS ARE UNNECESSARY

/IF MODE HAS BEEN SET ELSEWHERE

LAC (500020

CAM 5

JMP .-1

/N=13,F=16,A=0,M=100, EXECUTE

/ADDRESS SCAN READ. (FOR EXAMPLE: READ 12 SCALERS IN N=1,2,3)

/READ STARTING AT N=1,A=0, F=0,M=240

-14

DAC CNT

LAC (000000

DAC 10

LAC (400020

CAM 5

JMP .-1

LOOP LAC (400000

CAM 7

JMP .-1

CAM 1

JMP .-1

CAM 72

DAC 10

CAM 72

DAC 10

ISZ CNT

JMP LOOP

/N=1,A=0,F=0,M=240, SCAN

/EXECUTE BIT

/WAIT TILL SCAN FINISHED

/10 LSB

/5 MSB

.....

/CONTROL FUNCTION

/GENERATE 2 AND INHIBIT

LAC (400000

CAM 24

/ENABLE INTERRUPT AND INHIBIT

LAC (020000

CAM 24

/INTERRUPT PROCESSING
/SKIP CHAIN SHOULD INCLUDE

.
CAN 21
SKP
JMP CANAC
.

DISMIS (RESTORE REGISTERS)

ION
OBR
JMP 0

CANAC CAN 72 /SAVE REG. 3
OAC SAVR3
CAP 1
JMP -1 /UPDATE REG. 2, LAM STATUS
CAN 32 /READ 18 LSB INTO AC

.
TEST FOR MODULE CAUSING INTERRUPT
AND TAKE REQUIRED ACTION
EXIT THROUGH CAMDIS

CAMDIS LAC SAVR3
CAN 64 /RESTORE REG 3
JMP DISMIS

/ I/O TO OTHER REGISTERS, (I.E. REGISTERS 4 THROUGH 17)
/THESE OPERATIONS MUST EITHER BE DONE WITH THE INTERRUPT OFF
/OR REG. 2 MUST BE SAVED AND RESTORED IN THE INTERRUPT ROUTINES.

/AC OUTPUT

/LOAD CA FOR DCHB
LAC 11 /SET ADDRESS OF CA
CAN 24
LAC XCAT
CAN 44 /LOAD REG 11

/AC INPUT

/READ KC FOR DCHB
LAC 12
CAN 24
CAN 52 /KC TO AC

REFERENCES

1. "PDP-15 MACRO-15 Assembler Programmer's Reference Manual," Digital Equipment Corp. Report No. DEC-15-AMZA-D, Digital Equipment Corp., Maynard, Massachusetts 01754 (1969).
2. "CAMAC Organization of Multi-Crate System," TID-25876, AEC NIM Committee, ed. L. Costrell, United States Atomic Energy Commission (Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 20402 (1972).
3. "CAMAC A Modular Instrumentation System for Data Handling," TID-25875, AEC NIM Committee, ed. L. Costrell, United States Atomic Energy Commission (Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 20402 (1972).
4. "Supplementary Information on CAMAC Instrumentation System," TID-25877, AEC NIM Committee, ed. L. Costrell, United States Atomic Energy Commission (Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 20402 (1972).
5. A. L. Marusak, "FLOTSAM - A Two-Parameter On-line Data Acquisition Program for the PDP-7," ORNL-4299 (October 1968).
6. "Title FPOINT-EAE," copyright 1969, Digital Equipment Corporation, Maynard, Massachusetts 01754; Portions Reproduced with the Permission of Digital Equipment Corporation. All Rights Reserved.