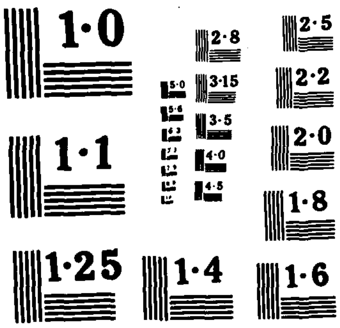
AD-A155 842   WHOLEBODY RADIATION COUNTING SYSTEM(U) ARMY MILITARY   1/2
PERSONNEL CENTER ALEXANDRIA VA G M FECHTER 01 MAY 85

UNCLASSIFIED   F/G 6/18   NL


1·0
2·8
2·5
3·15
2·2
5·0
5·6
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

DTIC FILE COPY

AD-A155 842

85 8 [illegible] 035

Wholebody Radiation Counting System

Captain Gary M. Fechter
HQDA, MILPERCEN(DAPC-OPA-E)
200 Stovall Street
Alexandria, VA 22332

Final Report 1 May 1985

A thesis submitted to Rensselaer Polytechnic Institute, Troy, New York, in partial fulfillment of the requirements for the degree of Master of Science.

WHOLEBODY RADIATION COUNTING SYSTEM

by

Gary M. Fechter

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Approved:

Robert M. Ryan
Thesis Co-Advisor

Willard A. Bryant
Thesis Co-Advisor

Rensselaer Polytechnic Institute
Troy, New York

May 1985

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to express his graditude to Helgeson Nuclear Services, Inc. for the donation of the basic components of the Do-It-Yourself Wholebody Counter. Without these items the Wholebody Counting System would not have been possible. Thanks are also extended to Mr. David Hollinger of Duffers Scientific, Inc. for his assistance in EPROM programming.

Special thanks are extended to Mr. Willard Bryant for his assistance in the design and construction of the Computer Interface Module.

ABSTRACT

The purpose of this research was to develop a system that would allow the Health Physics staff of the Nuclear Engineering Department of Rensselaer Polytechnic Institute to conduct wholebody radiation counting of all assigned personnel and to determine if a multiple crystal arrangement had any benefits over the single crystal system. The research had three main objectives:

1. To design and construct the electronic hardware necessary to collect data from a series of three NaI(Tl) scintilation crystals;

2. To develop the computer software necessary to interpret this data and calculate concentrations present within those individuals tested.

ABSTRACT

3. *To interface the Helgeson Wholebody Counter with* the Vax 11/750 Computer System.

This system includes a Texas Instruments TM990-101M Microcomputer which acts as the main control unit for the operator and as the interface with the Vax computer. Additionally, the final system included a computer interface and data acquisition module and it uses a Digital Equipment Corporation VAX 11/750 Computer System for a majority of the data manipulation. The crystal utilized are each three inches in diameter and three inches in thickness NaI(Tl) scintillation crystals. These crystals are placed in a carrier which moves over an individual from his feet to head over an eight minute period. *This system, as all* wholebody counting systems, is characterized by its ability to detect low levels of radionuclide concentration while maintaining adequate resolution to identify the exact isotopes present within the person being counted.

The system developed by the author utilizes a Helgeson Wholebody System with changes to the detectors, the computer systems, the data acquisition system and the analytical software utilized. The system utilizes a high speed data acquisition system to collect and digitize the detector output and interface the detectors with the microcomputer. This unit allows collection and interpretation of data

ABSTRACT

at rates in excess of 1 MHz. Additionally, the collection module is linked to the timing routine so that only live time is measured. This creates a system where the effects of dead time are reduced by timing only the interval between the detection of events. The system provides a sensitivity of less than 20 nanoCuries based on five counting cycles of one minute each. Additionally, the system has the capability to be expanded to provide more detailed information as to location rather than providing only total body burden.

CHAPTER 1

INTRODUCTION AND HISTORICAL BACKGROUND

The radionuclide concentration within humans is a concern to all workers whose duties bring them into contact with radioactive materials. These workers are constantly monitored for radiation exposure. These methods include personal dosimetry devices, air sampling devices and surface deposition sampling. These methods are designed to alert the Health Physics Staff to any potential exposure or uptake. These methods all concentrate on the worker's external environment. Once a person receives a reading above a certain level there is a need to verify the operation of protective devices and to insure that no internal contamination has occurred. This led to the development of systems that were able to measure the quantity of radioactive material within an individual. Three basic types of wholebody counting systems were developed in the 1950's and early 1960's. These were liquid scintillation detectors, the plastic scintillation detectors, and the Sodium Iodide detectors. These systems included the Los Alamos Model which had an array of photomultiplier tubes attached to a large volume of liquid scintillation fluid that was carried in a double walled cylinder that approximated four pi geometry. Another system was a steel room which was lined with lead and contained an 'Argonne Chair' and a single eight inch by four inch NaI(Tl) detector. A third system used large plastic scintillation detectors in an arrangement similar to the Los Alamos system./1

Each of these systems has demonstrated advantages and disadvantages. For the single instrument system the main advantage was the increased resolution. However, this was achieved with a corresponding decrease in the sensitivity of the system. The Las Alamos system demonstrated superior sensitivity, however, the system had poor resolution due to the large scintillation volume utilized. The plastic scintillators had a demonstrated performance in the middle

---

1/ Meneely, G.R. et. al., Measuring Gamma Activity with Wholebody Counters, Nucleonics, Vol.21, No 10, October 1963, p. 46.

of this range.

Another consideration with each of these system was the capital investment and the operation and maintenance costs of the system. The systems in use in the early 1960's were considered as major investments and the attitudes of scientist at the time were consistant with the following comment made at a Conference on Wholebody Radition Counting in 1963; "About $50,000 to $100,000 will be required for initial installation, to say nothing of the salaries of the team of scientists, engineers and technicians who will maintain and operate it."/2 These systems were difficult to maintain and normally required a dedicated team of technicians. This decreased the willingness of private facilities to invest in these systems. As the price of nuclear instrumention decreased so did the cost of these facilities and smaller more efficient systems were designed and constructed. These systems began to gain acceptance with the nuclear industry and work on their improvement continued. Additional savings were also achieved in the manpower reductions inherent to the more modern equipment. Another benefit was that a large staff was no longer

2/ Ibid p. 47.

to identify from which detector the pulse came. This would allow the operator to conduct more detailed mapping of the radionuclides by transmitting the detector number as well as the energy information for each pulse.

The initial phase of this program is to load the interrupt vectors. Each interrupt has specific memory locations which are searched for program information once the computer detects the interrupt. For example, the Level 7 interrupt uses memory locations FF0E, FF10, and FF12. These would be loaded with the following instructions if the counting routine for detector 1 was located in memory location F140:

```
FF0E     0420    Branch to location @>F140
FF10     F140
FF12     0380    Return to the main program.
```

The use of these branching vectors allows you to build various programs without the need to change the interrupt structure. In the current program each counting routine utilizes the same data storage area. This provides the user with the total observed counts from all three detectors.

value which represents to the proper number of cycles. The program increments a register [RO] everytime the interrupt is is generated. This is then compared to a given value to determine if the required counting time is complete. For this system the required time is one minute. Therefore, a value of 240 is used for comparision. This is the product of

$$1 \text{ min} \times \frac{60 \text{ sec}}{\text{min}} \times \frac{1 \text{ interval}}{0.25 \text{ sec}} = 240 \text{ intervals}$$

This value is then converted to hexidecimal and is entered as >FO. This program exits directly to the data transmission routine upon completion. This cycle is repeated five times during each individual count. This provides a total of five one minute counts separated by the data transfer time.

Data Collection Routine

This routine is based on the use of the seventeen levels of prioritize interrupt built into the Texas Instruments microcomputer. The data acquisition module is specifically designed and constructed to use these interrupts. The electronics of this module will be explained in detail in Chapter 4. The interrupts selected to be used by this module are Levels 7, 8, and 9. These allow the user than the single block used by the Vax routine. The initial block compares the character to a number symbol, , [ACSII value 23]. Once this is detected the computer exits the routine. This is used at the end of a session to terminate the program. The second comparision is to a dollar sign, $, [ASCII value 24]. Once this is detected the program jumps to the data collection routine. The third comparision is to a carriage return [ASCII value OD]. This is used to signal the end of a response other than counting or exiting the program. Once the carriage return is detected the routine would reestablish communications with the Vax and transmit the message. This cycle is continued until another routine is accessed or the program is terminated.

Timing Routine

The timing routine is based on the internal clock of the TM 990-101M computer. This clock uses a 3 MHz crystal to generate interrupts at specific intervals. The clock is internally wired to the level 3 interrupt. The interval used is based on the number of cycles requested. This program uses a value of 5B9F to establish an interval of 0.25 seconds. This is achieved since the TMS 9901 [Asynchronous Communications Controller] converts the value to a binary

Vax runs the programs as directed by the user. The next step is to load the base storage address. This is followed by reading the welcome message and the program menu and storing it in successive memory locations. The external programs are all constructed to transmit an asterisk to signal the end of a transmission. This was necessary so that the micrcomputer would have a signal to use to transfer control. The asterisk was selected since numerous transmissions included carriage returns and the output generated by different subroutines would contain end of transmission characters. These end of transmission signals were disabled so the microprocessor would react only to the internally generated signal. Once the asterisk is detected the program prints the end of message flags and transfers control to the Texas Instruments Microcomputer. The communications program module for the Texas Instruments Microcomputer is similar to the module for the Vax with the exception of using different protocol data values and a different CRU address. The data used by the Texas Instruments Microcomputer is 6200 and 0638. These correspond to a seven bit word, two stop bits, even parity and a baud rate of 110. The program then prints the message from the Vax and stores the response. The Texas Instruments Microcomputer program utilizes three decision blocks rather appropriate for a DEC compatable terminal. This is done through a series of commands and instructions to the TMS9902 [Programmable Systems Interface Module]. The program performs these functions by utilizing the following groups of commands. The initial five lines establish the Communications Register [CRU] address for the auxillary port. The next five lines initialize the TMS9902 to the proper protocol. This is done through the use of data values stored in the memory locations listed in the program. In the case of communicating with the Vax the values of A200 and 0034 are used. These are decoded by the TMS9902 as a protocol of a seven bit word, one stop bit, even parity, and a baud rate of 9600. The value for the baud rate at which the two computers will communicate was set at 9600 baud so that the time gap in data collection would be minimized during the data transfer intervals. The next step is to establish a link with the Vax so that the data manipulation programs can be interfaced. This is done by writing a message to the Vax. This message is the USERNAME and the PASSWORD required by the Vax operating system. These are stored by the program as a series of hexidecimal numbers as specified by the American Standard Computer Information Interface Council (ASCII). These values are written to the Vax and from this point until the session is completed the link to the Vax computer system. The primary port is connected to a teletype which acts as the operator's terminal. This teletype (TTY) is used to initialize the system, to start the program and to record the results of the counting session.

The microcomputer was programmed using the Texas Instruments Machine Language. A complete listing of this program is provided as Appendix A. This program can be divided into four specific functional modules. The modules are the interface routine, the timing routine, the data collection routine and the data transfer routine. The program was generated and tested in RAM and then recorded onto EPROM. This allows the user to simply apply power to the system, set the program counter, set the work space pointer and issue the execute command.

Interface Routine

The initial section of the program establishes communications through the auxillary RS-232-C Port (P3). This port is connected directly to the communications interface bus (DZ 11) of the Vax 11/750 and the Vax treats this input as it would any other terminal. As such the TM990-101M must be made to emulate terminal characteristics

CHAPTER 2

TEXAS INSTRUMENTS MICROPROCESSOR

The entire system is controlled at the remote site by the Texas Instruments Microprocessor. This counting system utilizes a TM990-101M Microcomputer to preform these functions. The microcomputer consists of a single printed circuit board type computer, which contains 1K of on board Random Access Memory (RAM) and 2K of on board Erasable Programmable Read Only Memory (EPROM). This was expanded through the use of the TM 990-201 Memory Expansion Board to provided an additional 4K of RAM and 4K of EPROM. The system has one 16 bit parallel I/O interface port and one modified EIA RS-232-C serial I/O interface port. These ports are of particular importance since the dual communication capability is necessary in our application. The primary communications port (P-2) allows this system to be controlled by an operator using a terminal at the remote site, while the serial interface port (P-3) provides the sixteen inch by four inch crystal is approximately $8000.00. While the total cost of three three inch crystals and the computer interface module is less than $1000.00. This savings in cost is of minimal importance unless one considers the loss of resolution versus the cost . In this system you are able to improve the degree of resolution while decreasing the cost of the system so that the additional expense in not required.

The final RPI system utilizes a Helgeson Wholebody System with changes to the detectors, the computer systems, the data acquisition system and the analytical software utilized. Three NaI(Tl) scintillation crystals, each three inches in diameter and one inch in thickness, are utilized to measure the radiation emitted. These crystals are placed in a carrier which moves from the individuals feet to head over an eight minute period. To insure adequate counting time the system is designed to pass over the individual twice. The system utilizes a high speed data acquisition system to collect and digitize the detector output and interface the detectors with the microcomputer. This high speed system is explained in detail in Chapter 4.

The multiple crystal arrangement provides the user with the potential of a a greater variety in modes of operation. These include the potential to map the location of the radionuclides rather than being provided only with a total body concentration. This mapping option has the potential to generate a grid type output defining areas of concentration rather than simply providing concentrations of the isotopes present. The multiple detector system also has the advantage of being much less expensive than the single crystal system. The cost of the single sixteen inch by

"Radiological Engineering". The majority of the students enrolled in this course have little to no knowledge of the Vax or Texas Instruments computer systems.

The use of multiple crystals prompted the need for a module to accept and interpret the data. The Computer interface module was designed and constructed for this system to accept pulses from the three photomultiplier tubes, shape these pulses and transmit the data to the microcomputer. This module is explained in detail in Chapter 4 and diagrams are provided as Appendix C. The module was also designed to generate the interrupt signals used by the data collection routines. This unit consists of a series of operational amplifiers used to amplify and shape the incoming pulse without distorting the energy information provided by it. This pulse is then input to an eight bit Analog to Digital convertor to generate the binary signal used by the computer. This module was constructed to utilize the interrupt structure of the Texas Instruments Microcomputer so that all interrupts are hardwired into the system. The system also provides a direct links to the data and address buses of the microcomputer and allows easy manipulation of data in both directions.

were placed in Erasable Read Only Memory units which allow the system to operate without the operator having to enter a lengthy machine language program. This program is discussed at length in Chapter 2 and a listing is provided as Appendix A. The Texas Instruments Microcomputer also provides all of the interface requirements between the two computer systems.

The Vax computer is used to execute a majority of the software used to determine concentrations detected. These programs are discussed at length in Chapter 3 and a complete listing is provided as Appendix B. These programs consist of a command program which intitiates the main FORTRAN program upon Login, a main FORTRAN program, and seven subroutines. These subroutines control the separate actions required to conduct the session. The seven subroutines include the counting routine, the measurement of background radiation levels, calibration of the system for energy information, plotting the data, printing the results and the instruction routines. This software package is a combination of command language and Fortran programs designed to accomplish the required tasks assuming the user has a minimum of knowledge regarding both computer systems. This allows the system to be used as part of a laboratory in the Nuclear Engineering Department's curriculum course

The Helgeson Wholebody Counter is of the shielded tub design and consists of a stainless steel tank similar to a bath tub. The tub is seven feet in length and thirty inches in width. The interior of the tub is lined with reactor grade lead bricks covered by a stainless steel shell. Normally the Helgeson system would include a single detector, the detector carrier, drive assembly and the computer systems necessary to control these units. The original design included a sixteen inches by sixteen inches by four inches thick NaI(Tl) crystal. The system donated to the Rensselaer Health Physics Laboratory included the tub, the drive units and the detector carrier assembly. In an attempt to determine if more accurate information as to nuclide location could be obtained from the system a series of three NaI(Tl) cystals were substituted for the single large crystal.

A Texas Instruments TM990-101M Microcomputer was selected to control the system. A teletype is connected to the microcomputer to act as the user's terminal. This terminal also provides a hardcopy record of all counting sessions. The teletype is connected to the primary port of the TM990-101M and the auxillary port is connected to the Vax computer. The Texas Instruments Microcomputer programs a four pi configuration to maintain the very high sensitivity and the Ge(Li) is used because of its ability to resolve individual species. This type of combined system allows you to maximize each detector and improve the overall operation of the system. These systems are still being designed with the goal of ensuring that workers have maintained their exposure as low as reasonablly acheivable.

The Rensselaer Polytechnic Institute System is designed around the Helegson Do It Yourself System. The person being counted lies in the tub and an operator initiates the counting routine. The system collects the pulses detected by three NaI(Tl) crystals and converts these pulses to levels usable by a microcomputer. This information is then transmitted to the Vax Computer which converts the data values to concentrations of specific radionuclides, subtracts the background radiation levels and generates a report which informs the individual of the isotopes present and the concentrations detected.

There are four main hardware components of the RPI system. These include the Helgeson Do-It-Yourself Wholebody Counter, the Texas Instruments TM990-101M Microcomputer, the Digital Equipment Corporation Vax 11/750 Computer and the computer interface module.

required to maintain and operate this type of system. The increased use of solid state and microcircuit technology has greatly increased the stability of these systems and has also reduced the level of knowledge required to operate these facilities. As advances in instrumentation continued a facility would be able to hire technicians to operate a system rather than engineers and scientists.

Today a large variety of systems are in use in research and commercial facilities. These systems utilize a variety of detectors and associated equipment to conduct these measurements. The equipment range from shielded rooms to portable facilities that can be made immediately available in the event of an accident. These system are still characterized by their ability to detect low levels of radionuclide concentration in humans, while maintaining adequate resolution to identify the isotopes present. The types of detectors in use today are the three mentioned earlier with the addition of Germanium or similar semiconductor type detectors. Some systems are now utilizing multiple detectors to overcome the shortcomings of a single detector system. One example of this type of system is a large volume liquid scintillator in conjunction with a Ge(Li) detector. The liquid detector is designed in

Once the vector locations are loaded with the proper instructions the program establishes the interrupts. The initial command of this section opens the CRU link by loading a value of >0100. This establishes communications with the TMS 9901. The next step is to set bit zero to a value of zero which enables the interrupt mask. The next step is to enable each desired levels of interrupt. This is done by setting the corresponding bit equal to one. This is accomplished by the following series of commands:

1D03	SBO 3	Enables Level 3 [Clock]
1D07	SBO 7	Enables Level 7 [Detector 1]
1D08	SBO 8	Enables Level 8 [Detector 2]
1D09	SBO 9	Enables Level 9 [Detector 3]

This routine starts the clock and waits for an interrupt to occur. This is accomplished through the use of a Jump Self command. This command locks the program in one location until the interrupts cause a branch. Once the branch occurs the other sections check to determine if all counting is complete. If the time is not complete the return command is issued. If the time is complete the program moves to the data transmission routine.

The next section of the program is the four interrupt programs. The timing program simply increments the register and returns to wait. Each detector has a specific program to collect data from that unit. All of these counting routines have similar commands, however, the addresses used vary. The first step is to disable all levels of interrupt. This insures that a second interrupt does not interfere with data collection. This also stops the clock so that the counting time is the actual time between reading events. This establishes a system where the dead time approaches zero. The next step is to read the value from the A/D converter. Each of the three A/D converters is selected through a different address. This allows you to collect from a specific detector. The addresses used are as follows:

Level 7	Detector 2	8002
Level 8	Detector 1	8000
Level 9	Detector 3	8004

Once these data values are read they are stored in successive memory locations. The next step is to check if the data buffer is able to accept additional data. Once the buffer is full the program branches to the data transmission routine and this routine determines the address to which the next branch should occur.

Data Transmission Routine

This routine is designed to convert the hexidecimal values collected by the counting routine into decimal numbers and transmit them to the Vax. This is accomplished by reestablishing communicaions with the Vax and then reading the first value. Each value is decremented as a series of registers are incremented. These registers are compared to ten at the end of each cycle. Once a register reaches ten the next register is incremented and the previous is reset to zero. This is continued until the original value equals zero. Once this occurs the digits are sent to the Vax followed by a carriage return to separate each data point. This continues until all values have been transmitted from the data buffer. After the first interval is complete the program returns to the collection routine. This continues for five intervals so that the data from five counting periods of one minute each will be transferred to the Vax. After the fifth transfer cycle the end of data statement is transmitted to the Vax. The Vax programs are designed to respond to any value greater than 300 as the end of data signal. This end of data flag is sent by loading the registers with a hexidecimal value of 3300 which is converted to a decimal value of 300 and this causes the

FORTRAN program to move to the next task. Following this the microcomputer program branches to the interface routine and waits for the results and a follow on message to be transmitted from the Vax.

These routines continues until to branch to each other depending on the instructions received until the program is terminated.

CHAPTER 3

VAX 11/750 COMPUTER PROGRAMS

The Wholebody Radiation Counting System utilizes a combination of programs to preform the required functions. The system is established so that it begins automatically upon LOGIN. This is accomplished through a LOGIN.COM file (Figure B-1), which is a program written in Vax Command language. The program is initiated by the Vax operating system and requires the user to have no knowledge or exprience with the Vax computer system. This was necessary since this system is to be used by the Health Physics Staff to moniter personnel who work in radiation areas and by students who are taking a course of instruction in Radiological Engineering. A majority of the users will be students and most have no experience with the Vax operating system. Through the use of a command file the user is not required to start the individual routines. The LOGIN.COM file will execute individual programs automatically. This includes the FORTRAN programs as well as the Command language program generated for this system. The program initially prints the welcome message and establishes a command to restart the program if an error in data entry is detected. This is used to save the student from having to continually reset the microcomputer and the detectors if an error in data format is made. Next, the program runs the FORTRAN program MAIN. This program will be described in detail below. The final section of the program signs the individual off of the Vax system at the completion of the counting session.

MAIN.Fortran

The program titled MAIN.Fortran is designed mainly as the flow control program for the system. The system utilizes a series of eleven subroutines with five linked data files. This type of structure is used to allow the user to have the option as to which subroutines are needed by him during each session. This includes the saving of calibration and background data so these routines are not required as part of each count. This is useful since the background and calibration routines require approximately twenty five minutes each and requring them to be run prior to every count would increase the counting time from twenty five minutes to approximately an hour and one half. The use of subroutines also aided in the development process since each module was debugged individually prior to being linked with MAIN. The FORTRAN program MAIN simply contains a series of print statements to print the menu, the reading of the user's option and a series of IF statements to direct you within the system. This series of print statements ends as all print routines do with the transmission of an asterisk. As described in Chapter 2 the asterisk is design as the end of message flag for the microprocessor routines and as such is added at each termination point. This cycle continues until the user enters option ten at which time control is returned to the LOGIN program.

Option 1-General Instructions

This option is designed to print the instruction for running the program. This set of instructions is designed for initial users and may be bypassed by reqular users. The routine used is also used in each of the subroutines to print the specific instructions for that section. The use of this type of message routine allows for rapid changes to the instructions and greatly simplified the programing. Changes are made by simply editing the data file and entering the new information in the exact format in which the instructions are to be printed. Since the message routine will duplicate the information in the data file this type of program eliminates the need for an extensive series of FORMAT statements. The alternative to this routine was to include a series of PRINT statements as part of each routine. Due to the length of the instructions and the potential for errors in changing a large number of FORMAT statements the message type of program was selected for all instruction modules.

Option 2- Count an Individual

The second option is the counting routine and is the largest module in the system. The counting routine is titled COUNT.Fortran and is provided as Figure B-10. This subroutine initially prints a menu that allows the user to receive the instructions, to count an individual or to return to the main program. The selection of the instruction option (0) branches you to the subroutine titled INSTCNT.Fortran. This subroutine is identical in design to INSTGEN.Fortran used for general instructions with the exception of the data file accessed. INSTCNT is provided as Figure B-5 and a listing of the current instructions are also provided in that figure.

Once the user selects the count an individual option (1) the program opens the data files needed by the counting program. These are the calibration values (CALIBRATE.Data) and the data file that is used to save the results of the current session (DATA.Data). The most current set of data is saved by this routine so that if the user desires a plot of the values he can obtain one through the plotting subroutine. The next stage of the process is to collect the personal data on the person being counted. The program reads values for name, Social Security account number, height, weight, sex and age. The program then prints a set of short instructions. These tell the person being counted to lie with his/her feet under the detectors and to start the carrier moving as a second person enters a "$" on the terminal. The signifigance of the dollar sign is to branch the microprocessor program to the counting subroutine as explained in Chapter 2.

The program then collects the data sent by the microprocessor. The microprocessor is sending ASCII values which the Vax interprets as integer values. These values are used to establish the total number of counts in each channel. The use of an eight bit analog to digital convertor in the interface module generates a data value in the range of zero to two hundred fifty five. This is used in the counting routine to establish 255 channels. Each channel is summed until a value in excess of 260 is sent by the microprocessor. As explained in Chapter 2 the value of 300 is used to signal the end of data transmission. Additionally, the program calculates the maximum value of any single channel during the collection phase. This value is not required for the counting routine, rather to provide data for the plotting routine. It is more convienent, however, to determine this value as part of this subroutine then during the plot subroutine.

The next step is to adjust for background levels. The background routine will be discussed in detail later in this chapter. The counting subroutine accesses the data file (BKG.Data) and reads the current background value for each channel. The measured values are then reduced by this amount. Since this is a measure of a random process there is a probability that actual number of counts minus background will be a negative value. However, this negative value has no physical signifigance and to avoid errors from negative values a check is conducted as part of the adjustment process. The program resets any negative value to a level of zero before continuing with the next data value.

The next phase is to determine in which channels the peaks occur. To determine if a peak exists the program compares a given value to its nearest six neighbors. This is done by determining if a given channel (X) is greater than the value of (X+1), (X+2) and (X+3). To reduce errors caused by statistical fluxuations the average of (X+1) and (X+2) and the average of (X+2) and (X+3) are compared to the value in channel (X) rather than comparing each value individually. If the value in channel (X) is greater than these two values it is then compared to similar averages for the values (X-1), (X-2), and (X-3). If a value meets the criteria of being greater than these four averages it is labeled as a peak. The program next sums the area under each peak. This is accomplished by adding the value of the peak channel to the three previous and the following three channels. The final stage of this peak search is to determine if the observed peak meets the statistical criteria of being within the 95% confidence interval. This is accomplished by calculating a variable named SIGNIF. This variable is equal to 1.96 times the square root of the sum of the square of the variance in the total area (SIGTOT) plus the square of the variance in the subtracted area (SIGUNDER). The value of 1.96 is used to provide the 95% interval. The area of the peak is then compared to this value. If the area of the peak is less than this criteria the peak is not recorded.

The program then adjusts these values for the effect of Compton scattering. Compton scattering is an interaction between an incident photon and an electron in the absorbing media. The photon transfers a portion of its energy to the electron and is scattered at some angle. The energy deposited is a function of this scattering angle and the resulting spectrum is a continuum with a maximum of 0.25 Mev less than the full energy of the incident photon.

$\Delta\lambda=(h/mc^2)(1 - \cos\theta)$ Note: θ = scattering angle

The probability of Compton interaction is a function of the media with which it is interacting and for NaI(Tl) this is approximately a linear function with energy. We expect to see a greater effect as the energy of the particle increases. Compton scattering is seen to the data program as an event where less than the full energy of the incident particle is deposited within the crystal. These incomplete interactions within the crystal result in counts being detected in the channels of energy lower than the full energy channel. This effect occurs with all photons, however, as explained above is more prominent with high energy photons. To correct for this effect the area must be reduced by the number of counts under this curve. The

Compton curve over short intervals approaches a linear relationship and in the small width that is being used by this subroutine that assumption does not introduce significant error. The area under the peak due to this additional curve is assumed to be equal to the width of the interval times the average height of the end points. The area of this rectangle is calculated by the program and its area is subtracted from the total area under the peak. Again, the random nature of nuclear events presents the potential for negative areas. This program resets any negative value to zero to reduce errors in future calculations.

The next section of the program searches for specific isotopes. These isotopes were selected based on their occurance in nature and based on the fact that they are of particular interest to radiation workers. This section initially calculates the energy of each channel. This is accomplished through the CS137 calibration factor and a constant to correct for an observed shift in the calculated energy of each channel. This shift is primarily dueto the calibration of the electronics of the data module. To determine the most accurate relationship between energy and channel number calibration runs were conducted with several sources (Cs137, Co60, Na22) and the results were used to develop the following equation:

Energy (Kev) = (PEAK * CALFAC) + 20.0

PEAK is defined as the channel number of the peak

CALFAC is defined as the Cs137 calibration factor

The search compares the energy of the peak to the energy of the characteristic gamma for each of the listed isotopes. The determination of isotopes greater than 1.02 Mev is based on the single and double escape peaks as well as the full energy peak. The program searches for the full energy peak and then determines if the single and double escape peaks are present and the ratio of these peaks. These escape peaks are a result of the process of pair production. If pair production occurs the incident photon deposits its energy in the crystal by producing an electron and a positron pair. Once the positron particle is slowed it under goes a process known as annihilation. If both of the annihilation photons are captured and absorbed within the crystal then the total energy of the initial photon is observed by the detector. These events are recorded as full energy events. In this process (Figure D-5 thru D-7) the positron particle gives up two photons of 0.511 Mev and the kinetic energy of the initial photon. If one annihilation photon escapes the crystal the event is recorded at full

The reset of the Flip-Flop is generated by the read operation and decoding of the read address. The decode device will be explained later in this chapter.

The second Harris HAI-5195 is used to further amplify the pulse, to invert the pulse and to alter its shape. The characteristic shape of this signal is shown in Figure 4-2. This amplifier is operated in the inverting mode so that negative pulse generated by the first amplifier will be converted to a positive signal in the range of zero to ten volts. The multiplication factor for this device is calculated as follows:

$$M = \frac{R1 + R2}{R1} = \frac{(20.0) + (30.0)}{(20.0)} = 2.5$$

Resistor values are in Kilo-Ohms

This provided a positive signal of approximately 3.5 volts for a Cesium 137 event from an incident voltage of approximately 1.41 volts.

$$V(out) = V(in) \times M = (1.41 \text{ volt}) \times 2.5 = 3.53 \text{ volts}$$

The ADC requires approximately one microsecond to complete a conversion and to insure that the actual value is measured, the peak value must remain present during the entire conversion period. This is accomplished by using an to zero the amplifier returns to saturation and the output level returns to 10 volts. The shape of this pulse and its relationship to the output pulse of the first stage can be seen in Figure 4-2. This signal is not at TTL level and must be sent through a voltage divider before it can be used in future stages. The signal from the voltage divider is a five volts positive signal when no pulse is present and once the peak occurs the signal drops to ground and remains there for approximately one microsecond. This transition is used to enable the Analog to Digital Converter (ADC) and to signal an interrupt to the microcomputer. The one microsecond pulse is long enough to be used as the signal to begin conversion for the ADC and this signal is, therefore, wired directly to pin 11 (Start Conversion) of that device.

The interrupt requires the pulse be present longer than the time achieved with this unit. Therefore, the output is used to trigger a J-K Flip-Flop which will hold the level until it is reset. This module uses a 74LS74 Flip-Flop to accomplish this task. The output of the voltage divider is wired to the Preset connection of the Flip-Flop (Pin 4). This causes the output at Q (Pin 6) to go to ground once the peak has been detected. The 74LS74 is designed to maintain the signal in the low position until the Flip-Flop is reset.

thirty five millivolts.

V(out) = V(in) x M = (0.035 volts) x 40.3 = 1.41 volt

The characteristic pulse shape for this amplifier is shown in Figure 4-1. The curves shown are for a Cesium 137 source and show that the OPAMP is achieving the multiplication of forty and that it is being operated in the non-inverting mode. This negative pulse is then sent to two separate amplifiers. Both of these devices are Harris Semiconductor operational amplifiers model HAI-5195-5. These again were selected for their response time and their ability to handle a high count rate. The first of these is used as a peak detector. This unit is designed to compare the incident pulse to a given reference voltage. For this module the reference voltages were plus and minus 5.2 volts. These were connected through a ten kilo-ohm potentiometer to allow for calibration of the system. By using a potentiometer at this point any DC offset generated by the first amplification could be corrected for at this point. This device is operated in saturation so that the normal output is 10 volts and once the peak occurs the output drops to ground. This occurs because the input signal is being compared to the reference voltage and once it drops below the peak value the unit is no longer in saturation and the output level drops to zero. Once the incident level returns initial amplification was to elevate the input signal to a voltage in the range of one to three volts. This was accomplished by using a Fairchild uA715 operational amplifier (OPAMP). This unit was selected for its ability to operate at very high rates with little distortion. The Fairchild OPAMP has a slew rate of 400 volts per microsecond and the ability to be compensated and adjusted through the use of external resistors and capacitors. This ability to be adjusted without the use of involved and difficult to construct circuits added to its usefulness. This compensation allows the removal of much of the oscillation that occurs with standard operational amplifiers. This oscillation is not acceptable since any distortion of the incident signal would result in the loss of the energy information carried on that pulse. This stability was provided by the Fairchild OPAMP and a multiplication factor (M) of approximately forty was obtained through proper resistor selection.

$$M = \frac{R1 + R2}{R1} = \frac{(.89) + (35.0)}{(.89)} = 40.3$$

Note: Resistor values are in Kilo-Ohms

This provided a signal of approximately 1.4 volts for a Cesium 137 event from an incident voltage of approximately

CHAPTER 4

COMPUTER INTERFACE MODULE

The design of the interface module was based on the need to amplify, shape, and digitize the output of the three NaI(Tl) crystals without losing the energy information provided by each pulse. The output of the three photomultiplier tubes was to be the input to the module and the desired output was a TTL level signal in the range acceptable to the Texas Instruments Microcomputer. The module was to be constructed on a prototyping board so that it would have direct access with the interface bus of the microcomputer.

This module is constructed to provide identical capabilities to each of the three detectors. This discussion will address a single unit with the understanding that except as noted the other two units are identical. The schematic of the first stage is provided as Figure C-1. The

Option 10- Terminate the Program

This option simply exits the FORTRAN program and returns you to the LOGIN program. Once you exit MAIN the command program will print the closing message and log the user off the Vax system.

increased width provides better resolution and a clearer plot. Additionally, the printing of three hundred lines at the slow printing speed of the teletype represents a wait of approximately 20 minutes. By using an alternate printer the user can continue with his session while the plot is being printed. This program then returns you to MAIN.

Option 7- Run a Supervisory Count

This subroutine is titled SUPCOUNT.Fortran and is provided as Figure B-13. This program is identical to the program COUNT.Fortran with one exception. The exception being that the counting routine requires you to have a peak greater than 1.96 times the square root of the area and the supervisory counting routine allows you to select any value for this parameter. This is useful during tests of the system when you wish to observe all peaks. This may also be used to increase the required significance by increasing this parameter to a value greater than 1.96, e.g. a value of 3.0 would be entered for 99% confidence.

Option 6- Plot the Current Data Values

This option calls the subroutine PLOT.Fortran which is provided as Figure B-12. This routine initially establishes the horizontal and vertical axis. This is done by loading a series of 300 lines with specific values. The first set of instructions create blank lines with a dash in column eleven to be used as the horizontal axis. Next the vertical axis and the vertical titles lines are generated. The next block of commands establish the horizontal headings and titles. This routine utilizes the current data set from the counting routine. The program accesses the data file and reads the value of YMAX and the values for each channel. These are used to calculate a new set of values equal to the current value in each channel as a percentage of YMAX. This is done to allow for uniform plotting of a data set that has a large variation in the number of counts in the peak channel. The values of YMAX can range from several counts to several thousand counts based on the strength of the radiation present. To avoid difficulties with the plotter the percentage is plotted rather than the actual peak height. Once these are calculated the plot is sent to the printer in the main computer room of the Nuclear Engineering Laboratory. This allows for a width of 135 characters versus the width of 70 characters on the teletype. This value (4) the program accesses the data file, reads the value, reads the new value from the terminal and then changes the value in the data storage file.

Option 5- Print the Current Calibration Values

This option calls a subroutine that is also used by the calibration routine. This printing routine is titled CALPRINT.Fortran and is provided as Figure B-9. This routine is a printing routine that provides the user with a list of all current calibration values or with the energy calibration value only. This is designed to pr vide students with the values used in the calculations so that they can independently verify concentrations present. Additionally, these may be needed if isotopes are present that are not currently being searched for by the counting routine. This routine begins by printing a menu and reading an option. If the user request only the energy factor (1) the program opens CALIBRATE.Data , reads the current value for CALFAC and prints that value. If the user requests the complete list (2) the program accesses the file to find each value and then it prints these values.

concentration calculations of the counting routine rely on the accuracy of these values and if they are changed without extensive recalibration of the system the concentration values generated would be worthless. Once the password has been verified the program branches to the subroutine titled CHANGES.Fortran. Upon completion of CHANGES the program returns you to the menu.

Changing Calibration Values

CHANGES.Fortran (Figure B-8) is designed to access and alter the contents of the calibration data set. The program initially prints a menu and reads your option. The options include entering a new value, printing current values, or changing current values. If the user request the enter new calibration value option (1) the program asks for the name of the new variable and the initial value. The name and value are then reprinted to allow the user to verify the accuracy of the variable name and its current value. If the user request to have all values printed (2) the program calls the subroutine CALPRINT. This program is used to print calibration values and is explained later in this chapter. If the user wishes to have only a single value printed (3) the program accesses the calibration file and prints that value. If the user wishes to change an existing

Option 4- Run the Calibration Routine

This option calls the subroutine titled CAL.Fortran (Figure B-6). This program has two separate subsections within it. It initially prints the menu and reads the user's option. The instruction option (0) is titled INSTCAL.Fortran (Figure B-7) and is of the same design as the three previous instruction routines. If the user selects the User Calibration Option (1) the program branches to a routine that receives the incoming data and calculates the peak channel. This is done in a manner similar to the counting routine's determination of YMAX. In this routine the only value of importance is the channel number of this maximum. Once the value of 300 is received the program calculates the value of CALFAC. This is obtained by the following equation:

CALFAC = 642.0 / (FLOAT(IPEAK))

The value of 642.0 is used to correct for the shift of the energy spectrum and the FLOAT command is used to allow you to divide by the integer value IPEAK. Once this value is calculated it is written to the Calibration data file (CALIBRATE.Data). The program then returns you to the menu. If the user selects the Supervisor Calibration Option (2) he is required to enter an additional password. This is installed to provide some protection to the system. The

Option 3- Run a Background Count

The background routine is titled BACKGND.Fortran and a listing is provided as Figure B-4. This subroutine is constructed very similar to the counting routine. It initially opens the data file BKG.Data and prints the menu. The initial option (0) is for instructions. This uses a subroutine titled INSTBKG.Fortran and has the same organization as the general instruction routine and the counting instruction routine. This subroutine is provided as Figure B-5 and the figure includes the current instructions. Once the person selects the background counting option (1) the short instructions are printed and data collection begins. This follows the same format as the counting routine in that all values received are used to increment the channel with the same number and an identical peak search routine is utilized. This program stores only the number of observed counts in each channel so that these values may be subtracted from the total counts during the counting subroutine. Once the value greater than 260 is received the subroutine prints the locations of the peaks, closes the data file and returns to MAIN.

percent of the time by beta emission with 94.6 percent populating the 0.662 Mev level of the Barium 137m. The Ba137m decays with a gamma 89.9 percent of the time. This combines to produce a final 0.662 Mev gamma abundance for Cesium 137 of 85 percent. Therefore, 37 disintegrations per second is an activity of 1.17 nanoCuries rather than the 1.0 nanoCuries 37 dps normally represents.

The next section prints the results of the session. It begins by printing all the personal data on the individual counted and then prints a listing of the energy of the observed peaks and the area under each of these peaks. Prior to printing a peak the program tests if the peak meets the statistical test of existance within the 95% confidence interval. This is measured by comparing the value for the area to 1.96 times the square root of the sum of the squares of that value. The program only prints those peaks that are greater than this value. The system also contains a counter that is incremented when any value is printed. This allows you to print a message if all concentrations are below the lower limit for the counting system. The final stage of the program closes the data files and returns you to MAIN.

energy minus 511 kev and this is known as the single escape peak. If both of the annihilation photons escape the crystal the event is recorded at full energy minus 1.022 Mev and is referred to as a double escape peak. One example of this is the observed spectrum of potassium 40. The characteristic gamma of K40 has an energy of 1.462 Mev. The single escape peak occurs at 951.0 Kev and the double escape occurs at 440.0 Kev.

The next step in the program calculates the nanoCuries of each radioisotope present based on the area calculated. This is done by multiplying the area by the calibration factor for that area based on each isotope. These factors account for a variety of factors in the system. Among these is that the efficiency of the detectors is not linear with energy. Because of this nonlinearity a single efficiency cannot be obtained for the system and used. Additionally, each isotope has a different yield and this results in different values of the calibration factor. One example of this is the decay of Cesium 137. Cesium 137 decays 100 amplifier that does not recover as quickly as the Fairchild OPAMP and by varying the resistor capacitor time constant for the system. This RC time constant was changed to provide a pulse that remained at its peak value for approximately seven microseconds. This signal is then sent through a 50 ohm resistor to the input connection of the ADC (Pin 20).

The Advanced Micro 8 bit Analog to Digital Converter AM-6108 was selected to digitize the pulses for this unit. The selection of this device was based on its extremely short conversion time, its ability to accept multiple input ranges and its compatability with microprocessors. The conversion time of the ADC is in many cases the limiting factor in determining data collection rate. With this unit the normal conversion time is in the range of 650 to 900 nanoseconds. This allows a counting rate in excess of one megahertz which is more than adequate for a wholebody system. This device is also manufactured in 10 bit and 12 bit models. These do not have the same conversion times as the 8 bit device. The 10 bit ADC has a mean conversion time of 4.5 microseconds and the 12 bit ADC has a mean conversion time of 7.0 microseconds. While the loss of speed would not have been of sufficient importance to justify using a device with less resolution it is a major design consideration. In this application the need for greater resolution was not justified. The output voltage of the second Harris amplifier is in the range of zero to 9.4 volts. This is for sources with a maximum gamma energy of 2.3 Mev. The use of the 8 bit ADC provides for a resolution of one part in 256 and this equates to 9.02 Kev per channel. The use of the 10 bit device would provide resolution of one part in 1024 and this would equate to 2.25 Kev per channel. This would be accomplished at an increase in cost from $23.56 per device to $127.45 per device. The use of the 12 bit ADC would provide a resolution of 0.56 Kev per channel at a cost of $389.70 per device. The decision as to which ADC to utilize was based on the cost per unit and on the fact that the detectors being used had a resolution in excess of 18 kev. The additional expense and decrease in counting rate were not justified since the resolution of the system is limited by that of the detectors. Had the system been based on a detector with a resolution of less than 10 kev the 10 or 12 bit ADC would have been installed.

The data lines of the AM 6108 are wired directly to the interface bus of the microcomputer. This is possible since the ADC provides tristate output to those data lines. These are connected to P-1 as shown in Figure C-2. The other connections to the ADC are to establish the range of input voltage, to provide the proper timing intervals and to control the read operation. For this application the voltage range of zero to ten volts was selected and the input was connected directly to Rin (Pin 20) and Roff (Pin 22) was not utilized.

The conversion routine requires a clock of 10 Megahertz to accomplish the conversion within one microsecond. The clock internal to the microprocessor has a maximum value of three Megahertz and could not be used for this module. This generated the need to construct a clock internal to the module. The wiring diagram for the clock is provided as Figure C-3. This clock utilizes two hex invertors and a 10 MHz crystal to produce the TTL levels needed. The hex invertors are contained in a single chip and to guarentee the ability to operate at 10 MHz a 74H04 was used in the clock circuit. This signal is sent to each of the three AM6108 devices (Pin 14).

Once the microcomputer detects the interrupt it branches to the address contained in a specific memory locations as explained in Chapter 2. The program then reads the contents of an address (8000, 8002, 8004) to obtain the value of the data point. These addresses are decoded by the 74LS139 which generates a signal when a specific address is requested. This signal serves two functions within the module. First, it resets the J-K Flip-Flop by activating the clear function (Pin 1). This resets Q to a high and disables the interrupt. Failure to reset this Flip-Flop would result in the microcomputer continually reading this address since the interrupt would appear to be signalling that another peak was present. The second function of this signal is to activate the Read function (Pin 13) on the appropriate converter.Once the Read connection becomes a low the ADC enables the tristate devices and the digital value of the analog pulse is seen on the data lines. If Read is not set low the data lines remain as zeroes. The wiring for the 74LS139 is shown in Figure C-4 and it operates on the following logic table:

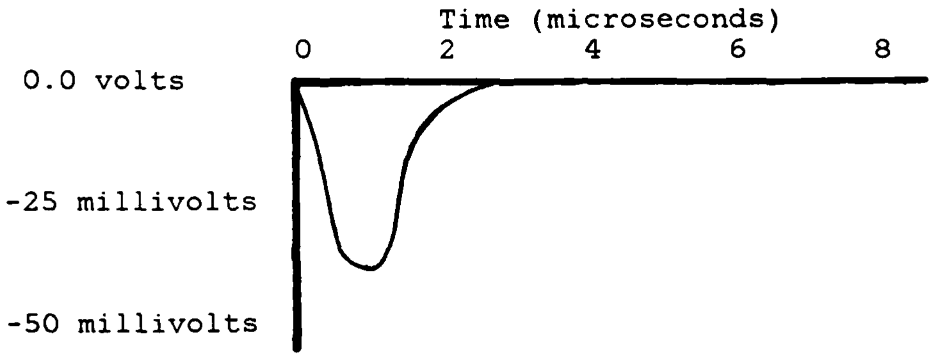
	DBIN	A0	A1	A13	A14	Y0	Y1	Y2
8000	L	H	L	L	L	L	H	H
8002	L	H	L	L	H	H	L	H
8004	L	H	L	H	L	H	H	L

From the logic table is can be seen that Y0 is used for the

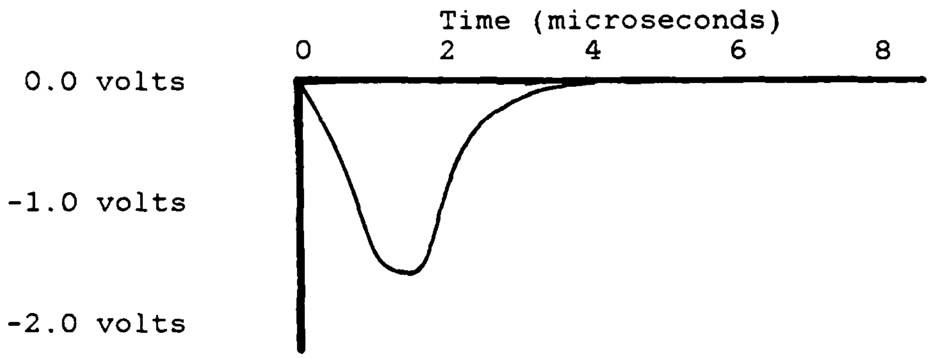
ADC connected the level seven interrupt, that Y1 is used for a level eight interrupt and that Y2 is used for a level nine interrupt. These signals are wired to the appropriate CLR connection of the three Flip-Flops so that the interrupt is disabled once the read is requested.

Figure 4-1

Characteristic Pulse Shapes for the
Fairchild uA-715 Amplifier



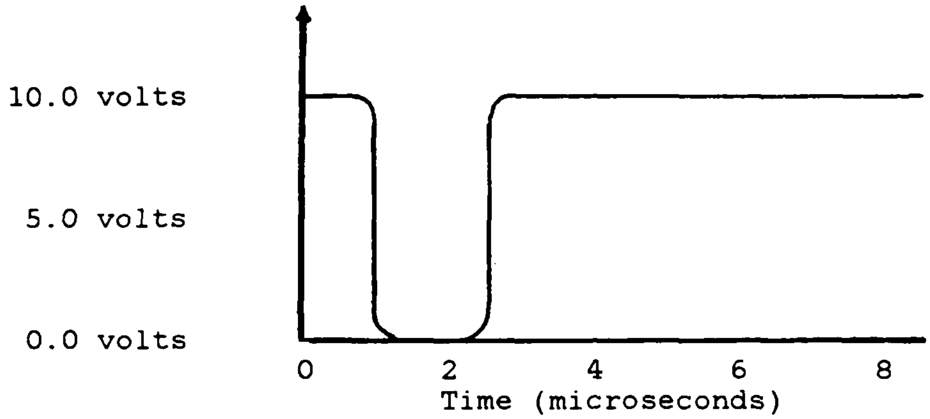
Input voltage to Fairchild Amplifier



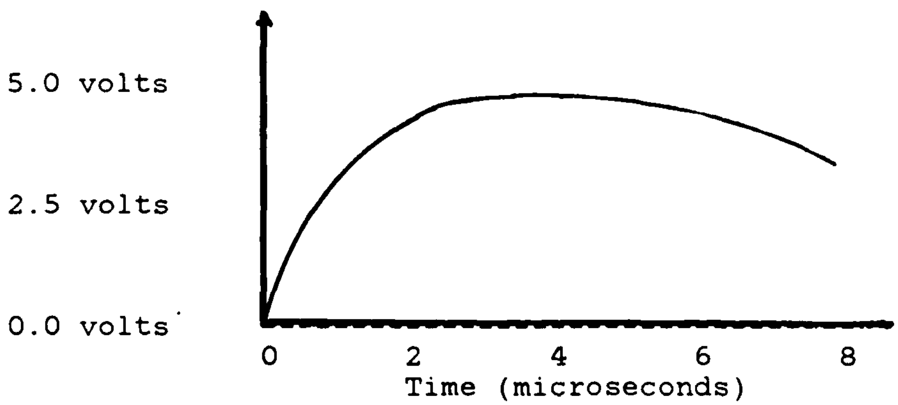
Output of Fairchild Amplifier

Figure 4-2

Characteristic Pulse Shapes from the
Harris HAI-5195-5 Amplifiers



Output of Peak Detector



Output of Second Harris Amplifier

CHAPTER 5

DISCUSSION AND CONCLUSIONS

The Rensselaer Wholebody Counting System allows the staff of the Nuclear Engineering Department to conduct wholebody scans of assigned personnel and to conduct a laboratory exercise for the Radiological Engineering students. The results of test scans utilizing calibrated sources showed that the system has a lower level of detection in the range of 20.0 nanocuries. This is based on five counting periods each of one minute. The system currently provides the user with a concentration based on the summation of all counts without regard to detector or location.

Appendix D contains the listing of the calibration runs conducted to verify the calibration factors. These are also provided to show the user the type of report expected from each type of source. The system continually identifies the presence of low concentrations of isotopes with low energy particles. This is due to the statistical nature of the process and due to incomplete collection within the crystals. Additionally, Appendix D contains the data sheets for the calibration sources utilized. These sources were provided by the United States Environmental Protection Agency Environmental Monitoring Systems Laboratory, Las Vegas, Nevada.

The data acquistions and interface module has become the only section of the system with recurring problems. Many of these are based on the relatively short lifetime of the Fairchild operational amplifier. These devices have been out of production until recently due to problems in the manufacturing process. It appears that the initial shipments of these amplifiers still demonstrate the problems that forced the cancellation of shipments for approximately one year. The replacement of one of these chips also requires that the operator recalibrate the electronics of the system and rezero the pulse height detector. This is accomplished by adjusting the ten kilo-ohm potentiometer for that module. This short lifetime could also be a function of high current spikes caused by stray capacitance within the system. In any high speed switching operation such as these amplifiers the peak currents can be as large as ten times the operating currents. This is a function of the layout of the module and any capacitance introduced between units. The unit was constructed to provide the shortest distance between units, however, the distance to this amplifier was not specifically addressed. This layout could cause the transient currents which would shorten the life of the device. It is recommended that any modification of this curcuit specifically address the problem of stray capacitance and actions be taken to reduce or eliminate this effect.

This interface unit does allows for significant expansion of the system. These improvements and modifications include the incorporation of a motor drive control into the interface module. The drive motors on the detector carriers currently have no way of externally controlling the units. They simply move continuously from one end to the other until a reset switch is physically activated. The potential exists to modify the motor drive controls by incorporating the directional relays into the interrupt structure of the microcomputer. In this way the carrier would move for one minute during the initial counting cycle. Once the timing routine transfers to the data transmission routine the intitial command would stop the movement of the detectors. This initial data set would then be from the first nine inches of the person. The Vax software could be modified to accept this data in one minute blocks and calculate concentrations in that region as well as in the entire person.

A modification of the machine language program would also allow the user to be provided with the detector number from which data point was received. This could be accomplished by changing the counting routine so that each detector utilized a separate block of memory, rather than sharing a single block of memory as is currently being done. This would also require a modification to the Vax software so that each minute of data would be treated as a separate data set. It is also recommended that all calculations be conducted using the summation of the three data sets. This is recommended since the number of pulses obtained by the separate detectors may not be sufficient to satisfy the statistical check, while the sum of the three may meet the criteria. This would allow for mapping of concentrations perpendicular to the direction of travel. This could then be combined with the results of the segmented count described above to generate a mesh of concentration values.

By adding both of these modifications a series of the partial concentrations would be generated. These would then be summed to provide a more accurate total concentration and additional interim results could be printed. This mesh would be very beneficial in that it could determine if the concentrations were localized or if they must be treated as a total body burden.

The resolution of the system could also be improved by changing the type of detectors being used. The interface module requires an electronic signal in the range of ten to one hundred millivolts to operate correctly. So any detector could be substituted for one or for all three of the detectors to increase the resolution. One suggestion would be to replace the center detector with a Ge(Li) to improve the resolution, while, maintaining the NaI(Tl) crystals as the outer detectors. This would provide results similar to other combined systems and would require only modifications to the software.

This system has demonstrated that a wholebody counting system can be constructed using a series of smaller crystals to replace a single large crystal. This conversion does not require the system to count for a longer period than is required for the single crystal. The original Helegeson system counted for an eight minute period as the detector carrier moved from one end to the other, and in this system the counting periods is only five minutes. The actual session time is increased in the RPI system due the data transfer routine. This transfer routine converts all data values to decimal numbers and transmits them to the Vax. This requires approximately .25 seconds per data value. This results in a period of three to four minutes between the counting cycles. This long delay could be eliminated by expanding the memory of the microcomputer. If the data buffer were made sufficiently large the person being counted would only have to lie in the tub for eight minutes and eight minutes worth of data could be collected. If this amount of memory were added, the counting routine could be modified to use succesive blocks for each minute of collection. This would allow for the grid mapping described earlier to be accomplished without the use of motor control units.

Another problem was encountered due to the limited address decoded done by the 74LS139 multiplexing unit. This device uses only address lines A0, A1, A13, and A14. This results in the need to restrict the use of address locations 8000 to CFFF. This is caused by the similarity in the four lines used among this block of addresses. By only using four address lines the multiplexing unit will actual respond to any address that will match the following binary pattern:

	A0	A1	A2 - A12	A13	A14	A15
"8000"	1	0	X	0	0	X
"8002"	1	0	X	0	1	X
"8004"	1	0	X	1	0	X

In the figure above the X indicates that any value may be entered without changing the result. From this you can see that the majority of the block from 8000 to CFFF would cause the enabling of the output tristates of the three converters. This causes problems in that the values on the data lines are read by the computers processing unit and it attempts to treat these entries as commands. These values rarely have any similarity to the commands of the program, but they can be interpreted as actions for the computer. This caused significant problems during the testing phase when the program was loaded into address locations B000 to B350. The result was a program that branched in a random manner and accomplished none of the required tasks. If this block of memory is to be used to allow for a single storage cycle it is recommended that the address decoding be expanded to include all 16 address lines.

```
2F06            XOP R6,12          :Prints the carriage return

0460            B,@>0818           :Branches to address  0818
0818                               :which is the read section
                                   :for  the  main  program.
```

The following is the data used by the TI990-101M to run the program. This is currently being stored in memory locations 0D00 to 0D1A.

```
A200    :Establishes communications protocol as a
        :7 bit word,  1 stop bit and  even parity.

0034    :Establishes a baud rate of 9600

6200    :Establishes communications protocol as a
        :7 bit word, 2 stop bits and  even parity.

0638    :Establishes a baud rate of 110
```

ASCII values for the Signon message.

```
0D0D    :<CR> <CR>
0D0D    :<CR> <CR>
5748    :WHOLEBODY<CR>
4F4C
4542
4F44
590D
4741    :GARY<CR>
5259
0D00
0700
```

```
16F2             JNE NXX         :If R8 is less than
                                 :9 this jumps to NX

0208             LI R8,>3000     :If R8 is greater than
3000                             :9 this resets R8 to 0

A1C5             A R5,R7         :Increases R7 by 1

10EE             JMP NXX         :Goes to NXX

2F07     OT      XOP R7,12       :Prints the  left  digit
2F08             XOP R8,12       :Prints the center digit
2F09             XOP R9,12       :Prints the  right digit
2F06             XOP R6,12       :Prints the carriage return

8102             C R4,R2         :Checks if all data has been
                                 :transfered and stops at the
                                 :last address  used  by  the
                                 :counting routine

16E1             JNE RDX         :If not branches to the read
```

This section determines if the five counting cycles are complete. It branches to the restart point if all five are not complete and prints a value greater than 300 when all are completed. This value is used as the end of data flag by the VAX program.

```
058B             INC RB          :This  increments the  cycle
                                 :counter following each data
                                 :transfer routine.

028B             CI RB,>5        :This compares the counter to
0005                             :five and exits when a  value
1302             JEQ FN          :of five is obtained.  If the
0460             B @>08A6        :value is less than five  the
08A6                             :program returns  to  restart
                                 :the counting routine.

0207     FN      LI R7,>3300     :Loads  the  ASCII  value
3300                             :of 3 into the left digit

2F07             XOP R7,12       :Prints the  left  digit
2F08             XOP R8,12       :Prints the center digit
2F09             XOP R9,12       :Prints the  right digit
```

The following is the program segment to convert a Hexidecimal value to its ASCII equivilent.

```
0204            LI R4,>D000       :Loads the base address
D000                              :for  the data storage.

0205            LI R5,>0100       :Loads the increment value
0100

0206            LI R6,>D00        :Loads the ASCII value of a
0D00                              :carriage return

0207    RDX     LI R7,>3000       :Loads the ASCII value of 0
3000

0208            LI R8,>3000       :Loads the ASCII value of 0
3000

0209            LI R9,>3000       :Loads the ASCII value of 0
3000

C2B4            MOV *R4+,R10      :Moves the  value from the
                                  :the memory location in R4
                                  :into R10

028A    NXX     CI R10,0          :Compares this value to 0
0000

130F            JEQ OT            :Move to OT if R10 equals 0

6285            S R5,R10          :Decreases R10 by 1
A245            A R5,R9           :Increases R9  by 1

0289            CI R9,>3A00       :Checks if R9 is
3A00                              :greater  than 9

16F8            JNE NXX           :If less than 9 goes to NX

0209            LI R9,>3000       :If R9 is greater than
3000                              :this resets  R9  to 0

A205            A R5,R8           :Increases R8 by 1

0288            CI R8,>3A00       :Checks if R8 is
3A00                              :greater  than 9
```

```
0580                INC R0          :This inrements R0  by  one
020C                LI R12,>100     :and reestablishes the  CRU
0100                                :operations to continue the
0201                LI R1,>5B9F     :timing program. This loads
5B9F                                :a value of 5B9F  which  is
                                    :used to  provide  the 0.25
                                    : seconds of  time interval.

33C1                LDCR R1,15      :This  enables  the   counter
1E00                SBZ 0           :and establishes the interrupt
                                    :mode on the TMS 9901.

1D03                SBO 3           :Enables the level 3 Mask.

0300                LIMI 3          :Enables Level 3 on the
0003                                :TMS 9901.

0380                RTWP            :This returns  and waits
                                    :for  another  interrupt.


0460      PT9       B @>0B00        :When the timing interval
0B00                                :has  been  completed the
                                    :program  branches to the
                                    :data transfer subroutine.
```

The next section converts the Hexidecimal information to ASCII characters and writes these values to the VAX.

Establishes communications with the VAX

```
020C                LI R12,>180     :This  establishes  the
0180                                :communications with P3
C80C                MOV R12,@>FFDE
FFDE
1D1F                SBO 31

3220                LDCR @>0D00,8   :This loads the
0D00                                :VAX  protocol.
1E0D                SBZ 13

3320                LDCR @>0D02,12  :This loads  the baud
0D02                                :rate used by the VAX
```

Level 9

```
F600            DATA >F600,STRT4 :Loads work pointer and
096C                             :the   program  counter.

0300    STRT4   LIMI 0          :Disables all interrupts
0000


COEO            MOV @>8004,R3   :Stores the value from
8002                            :the  A/D  converter

0283            CI R3,>0        :Compares the data value to
0000                            zero.  If the value equals
1601            JNE SV          :zero it is not saved.   If
0380            RTWP            :it does not equal zero  it
CC83            MOV R3,*R2+     :is stored in the next data
                                :location.

0282            CI R2,>F000     :Compares  storage address
F000                            :to the  maximum and exits
1301            JEQ PT5         :when all memory is filled
0380            RTWP            :Returns and waits for the
                                :next interrupt.

0460    PT5     B @>0B00        :When the  buffer  is full
0B00                            :the  program branches  to
                                :the data transfer routine.
```

Level 3 Clock

```
F600            DATA >F600,STRT5 :Loads work pointer and
098E                             :the   program  counter.

0300    STRT5   LIMI 0          :Disables all interrupts
0000

0280            CI R0,>F0       :Compares the value in R0
00F0                            :to  240  to determine if
                                :one minute  has  elapsed.

130B            JEQ PT9         :If it is equal it exits.
```

Level 7

```
F600            DATA >F600,STRT2  :Loads work pointer and
0944                              :the   program  counter.

0300   STRT2    LIMI 0           :Disables all interrupts
0000

C0E0            MOV @>8004,R3    :Stores the value from
8004                             :the  A/D  converter

0283            CI R3,>0         :Compares the data value to
0000                             :zero.  If the value equals
1601            JNE SV1          :zero it is not saved.   If
0380            RTWP             :it does not equal zero  it
CC83   SV1      MOV R3,*R2+      :is stored in the next data
                                 :location.
0282            CI R2,>F500      :Compares  storage address
F500                             :to the  maximum and exits
1301            JEQ PT2          :when all memory is filled
0380            RTWP             :Returns and waits for the
                                 :next interrupt.

0460   PT2      B @>0B00         :When  the  buffer is full
0B00                             :the  program  branches to
                                 :the data transfer routine.
```

Level 8

```
F600            DATA >F600,STRT3  :Loads work pointer and
0958                              :the   program  counter.

0380   STRT3    RTWP             :This interrupt is for future
                                 :expansion and is   currently
                                 :not being used.  The program
                                 :sends you back  to  wait  if
                                 :this   level   is  activated.
```

```
0201            LI R1,>380       :This loads a RTWP command
0380

C801            MOV R1,@>FFAE    :This moves the command
FFAE                             :to the proper location
```

This loads the base address to data storage.

```
0201            LI R1,>D000      :This loads  the base
D000                             :address for detector
C801            MOV R1,@>F604    :One into Register 2.
F604

0201            LI R1,>E000      :This  loads the base
D000                             :address for detector
C801            MOV R1,@>F606    :Two into Register 3.
F606
```

This establishes the interrupts and starts the process.

```
020C            LI R12,>100      :Establishes the CRU base
0100                             :address to the interrupt.

1E00            SBZ 0            :Enables the Interrupt Mask
1D03            SBO 3            :Enables Interrupt 3
1D07            SBO 7            :Enables Interrupt 7
1D09            SBO 9            :Enables Interrupt 9

0300            LIMI 9           :Sets the lowest active
0009                             :interrupt as a level 9

0201            LI R1,>3         :Places 2 ones in R1
0003

33C1            LDCR R1,15       :Sets up 9901

10FF    INLP    JMP INLP         :Waits for an interrupt
```

The next section of the program begins in memory location 0940 and is the individual program segments that each interrupt will branch to during the counting phase.

```
0380

C801          MOV R1,@>FF06    :This moves the command
FF06                           :to the proper location
```

Interrupt level 9

```
0201          LI R1,>420       :This loads a BLWP command
0420

C801          MOV R1,@>FEF6    :This moves the command
FEF6                           :to the proper location

0201          LI R1,>0968      :This loads the  branch
0968                           :point (vector) address

C801          MOV R1,@>FEF8    :This moves the command
FEF8                           :to the proper location

0201          LI R1,>380       :This loads a RTWP command
0380

C801          MOV R1,@>FEFA    :This moves the command
FEFA                           :to the proper location
```

Interrupt level 3 (Clock)

```
04E0          CLR R0           :This  clears  the  counter
F600                           :used by the timing routine

0201          LI R1,>420       :This loads a BLWP command
0420

C801          MOV R1,@>FFAA    :This moves the command
FFAA                           :to the proper location

0201          LI R1,>098A      :This loads the  branch
098A                           :point (vector) address.

C801          MOV R1,@>FFAC    :This moves the command
FFAC                           :to the proper location
```

This section sets up the timer and the other interrupts. The initial section loads all the interrupt vectoring information.

Interrupt level 7

```
04E0    ST6     CLR RB            :This clears the interval
F616                              :counter register.

0201            LI R1,>420        :This loads a BLWP command
0420

C801            MOV R1,@>FF0E     :This moves the command
FF0E                              :to the proper location

0201            LI R1,>0940       :This loads the  branch
0940                              :point (vector) address

C801            MOV R1,@>FF10     :This moves the command
FF10                              :to the proper location

0201            LI R1,>380        :This loads a RTWP command
0380

C801            MOV R1,@>FF12     :This moves the command
FF12                              :to the proper location
```

Interrupt level 8

```
0201            LI R1,>420        :This loads a BLWP command
0420

C801            MOV R1,@>FF02     :This moves the command
FF02                              :to the proper location

0201            LI R1,>0962       :This loads the  branch
0962                              :point (vector) address

C801            MOV R1,@>FF04     :This moves the command
FF04                              :to the proper location

0201            LI R1,>380        :This loads a RTWP command
```

```
0203            LI R3,>700        :Loads the end of
0700                              :message  flags.
CC43            MOV R3,*R1+
```

Reestablishes communication with the Auxilary port.

```
020C            LI R12,>180       :This  establishes  the
0180                              :communications with P3
C80C            MOV R12,@>FFDE
FFDE
1D1F            SBO 31

3220            LDCR @>0D00,8     :This loads the VAX
0D00                              :protocol.
1E0D            SBZ 13

3320            LDCR @>0D02,12    :This  loads  the baud
0D02                              :rate used by the VAX.

2FA0            XOP @>D000,14     :This writes a message
D000                              :from  the  teletype.

0201    VR1     LI R1,>D000       :This reads the characters
D000                              :from  the  VAX  and moves
2F43    NX1     XOP R3,13         :these characters  to  the
DC43            MOVB R3,*R1+      :next  memory  location.

0283            CI R3,>2A00       :This searches for an  *
2A00                              :and returns to the read
16FB            JNE NX1           :if one is not detected.

0203            LI R3,>D0A        :This loads the carriage
0D0A                              :return,  the  line feed
CC43            MOV R3,*R1+       :and the end of  message
0203            LI R3,>0700       :flag.
0700
CC43            MOV R3,*R1+

10CA            JMP L1            :This returns and writes
                                  :the message  to Port 2.

0460    ST      B,@>80            :Ends the program once a
0080                              : # symbol  is  detected.
```

```
0203          LI R3,>D0A        :This loads the carriage
0D0A                            :return, a the line feed
CC43          MOV R3,*R1+       :and the end  of message
0203          LI R3,>0700       :flag.
0700
CC43          MOV R3,*R1+
```

This establishes communications with the Teletype.

```
020C          LI R12,>80        :This establishes the
0080                            :base address for the
C80C          MOV R12,@>FFDE    :teletype.
FFDE
1D1F          SBO 31

3220          LDCR @>0D04,8     :This loads the teletype
0D04                            :protocol
1E0D          SBZ 13

3320          LDCR @>0D06,12    :This loads the teletype
0D06                            :baud rate

2FA0          XOP @>D000,14     :This reads the message
D000                            :sent by the VAX

0201          LI R1,>D000       :Loads  the base address
D000                            :for the message storage

2EC3    RD    XOP R3,11         :This reads and stores the
DC43          MOVB R3,*R1+      :characters from  Port  2.

0283          CI R3,>2300       :This  searches for a  #
2300                            :and  moves to stop once
1323          JEQ ST            :this symbol is detected

0283          CI R3,>2400       :This searches for a $
2400                            :and exits the program
1322          JEQ ST6           :once one is detected.

0283          CI R3,>D00        :Compares  to  a carriage
0D00                            :return and jumps to read
16F5          JNE RD            :the next character  if a
                                :carriage return  is  not
                                :entered.
```

APPENDIX A

TEXAS INSTRUMENTS MICROCOMPUTER PROGRAMS

The following is a listing of the machine language program that is used by the Texas Instrument's Microcomputer.

This section establishes the link with the VAX

```
020C    START   LI R12,>180       :This  establishes  the
0180                              :communications with P3
C80C            MOV R12,@>FFDE
FFDE
1D1F            SBO 31

3220            LDCR @>0D00,8     :This statememt loads the
0D00                              :protocol used by the VAX
1E0D            SBZ 13

3320            LDCR @>0D02,12    :This loads the baud rate
0D02                              :used  by  the  VAX

2FA0            XOP @>0D08,14     :This writes the LOGIN
0D08                              :message  to  the  VAX

0201    VR      LI R1,>D000       :This reads  characters
D000                              :from the VAX and moves
2F43    NX      XOP R3,13         :them  into  successive
DC43            MOVB R3,*R1+      :locations.

0283            CI R3,>2A00       :This  searches  for an  *
2A00                              :and  returns  to the read
16FB            JNE NX            :line if the character was
                                  :not  an  asterick.
```

LITERATURE CITED

9. Koslalo, M.J., et. al., An Accurate Gamma Spectrum Analysis Method for Minicomputers, Nuclear Instrumentation and Methods, Vol 190, 1981, pp 89-99.
10. Routti, J.T. and S.G. Prussin, Photopeak Method of Gamma-Ray Spectra from Semiconductor Detectors, Nuclear Instrumentation and Methods, Vol 72, 1969, pp 125-142.

11. Texas Instruments Engineering Staff, TM990 Introduction to Microprocessors, Hardware and Software, Texas Instruments Inc., October 1979.

12. Texas Instruments Engineering Staff, TM990/101M Microcomputer User's Guide, Texas Instruments Inc., May 1978.

13. Texas Instruments Engineering Staff, TM990/201 Memory Expansion Board, Texas Instruments Inc., Jan 78.

14. Texas Instruments Engineering Staff, TM990/302 Hardware User's Guide, Texas Instruments Inc., Jan 79.

15. Texas Instruments Engineering Staff, TM990/302 Software Development Board, Texas Instruments Inc., May 1979.

16. Wobschall, Darold, Circuit Design for Electronic Instrumentation, Mc Graw Hill Inc., New York, Copyright 1979.

LITERATURE CITED

1. Cember, Herman, Introduction to Health Physics, Pergamon Press, Elmsford, New York, Copyright 1983.

2. Chen, N.S., et. al., Application of a Conincidence Counting Technique in a Fixed Geometry Whole Body Counter, International Journal of Nuclear Medicine and Biology, Vol 1, 1974, pp 175-180.

3. Chen, N. S. and S. H. Cohn, An Improved Radionuclide Distribution Profile from a Fixed Array Whole Body Counter Using a Mathematical Analysis Technique, International Journal of Nuclear Medicine and Biology, Vol 1, 1974, pp 169-173.

4. Cirovic, Michael M., Basic Electronics: Devices, Circuits, and Systems, Reston Publishing, Reston, Virginia, Copyright 1979.

5. Cohn, S.H., et. al., A Whole Body Counter with an Invarient Response to Radionuclide Distribution and Body Size, Physical Medicine and Biology, Vol 14, No 4, 1969, pp 645-658.

6. Fairchild Camera and Instrument Corporation, 1982 Linear Divisions Products, Fairchild Corporation, Mountain View, California, Copyright 1982.

7. Freidlander, Gerhardt, et. al., Nuclear and Radiochemistry, Wiley and Company Publishers, New York, Copyright 1981.

8. Knoll, Glenn F., Radiation Detection and Measurement, John Wiley and Son, New York, Copyright 1979.

APPENDIX B

VAX 11/750 COMPUTER PROGRAMS

The following are the program used by the Vax Computer to preform all data manipulation for the Wholebody System.

Figure B-1

LOGIN Command Program

```
!
!       This establishes the required characteristics
!       for the terminal.
!
$SET TERMINAL/NOTYPE_AHEAD/WIDTH=68
!
!       This prints the Welcome message
!
$TYPE SYS$INPUT:

WELCOME TO THE RENSSELAER POLYTECHNIC INSTITUTE
    WHOLEBODY RADIATION COUNTING PROGRAM

THIS  PROGRAM  IS  DESIGNED  TO  MEASURE  RADIONUCLIDE
CONCENTRATIONS WITHIN  A  SUBJECT  INDIVIDUAL  UTILIZING
THE HELGESON WHOLEBODY COUNTER AND THE TEXAS INSTRUMENTS
TM990/101M MICROCOMPUTER.
```

Figure B-1 (continued)

```
THIS PROGRAM WAS PREPARED AS PART OF WORK  SPONSORED BY
RENSSELAER POLYTECHNIC INSTITUTE.  NEITHER  RPI NOR ANY
AGENCY THEREOF,  OR ANY  EMPLOYEES  MAKE  ANY WARRANTY,
EXPRESSED OR IMPLIED,  OR  ASSUMES  ANY LEGAL LIABILITY
OR RESPONSIBILITY  FOR  ANY  THIRD  PARTY'S USE OF THIS
PROGRAM.

COPYWRITED  1984
!
!       This runs the FORTRAN program MAIN and
!       establishes the return loop if an error
!       occurs during the running of the program.
!
$START:
$ON ERROR THEN GOTO BYPASS
$RUN [WHOLEBODY]MAIN
$GOTO END
!
!       This prints the error message and restarts
!       the program.
!
$BYPASS:
$TYPE SYS$INPUT:
    YOU HAVE MADE AN ERROR IN DATA ENTRY

PLEASE, CONSULT YOUR LAB ASSISTANT OR CHECK THE
INSTRUCTIONS IF YOU ARE NOT FAMILIAR  WITH  THE
REQUIRED FORM OF THE DATA.

   THE PROGRAM WILL RESTART AUTOMATICALLY.
$GOTO START
!
!       This terminates the program and prints the
!       closing message.
!
$END:
$TYPE SYS$INPUT:
   THIS CONCLUDES THE WHOLEBODY COUNTING SESSION.

TO RESTART THE PROGRAM YOU MUST INITIALIZE THE SYSTEM
       AS DESCRIBED IN YOUR LAB HANDOUT SHEET.

QUESTIONS SHOULD BE ADDRESSED TO YOUR LAB ASSISTANT*
$LOGOUT
```

Figure B-2

FORTRAN Program MAIN

```
      LOGICAL*1 OPTION
C
C     THIS PRINTS THE MENU
C
 1    PRINT 199
      PRINT 200
      PRINT 201
      PRINT 202
      PRINT 203
      PRINT 204
      PRINT 205
      PRINT 206
      PRINT 207
      PRINT 208
      PRINT 210
      PRINT 211
      PRINT 212
C
C     THIS SECTION READS THE OPTION AND CALLS THE
C     APPROPRIATE SUBROUTINE TO EXECUTE THAT OPTION
C
      READ(6,250),OPTION
      IF(OPTION.EQ.1)CALL INSTGEN(*1)
      IF(OPTION.EQ.2)CALL COUNT(*1)
      IF(OPTION.EQ.3)CALL BACKGND(*1)
      IF(OPTION.EQ.4)CALL CAL(*1)
      IF(OPTION.EQ.5)CALL CALPRINT(*1)
      IF(OPTION.EQ.6)CALL PLOT(*1)
      IF(OPTION.EQ.7)CALL SUPCOUNT(*1)
      IF(OPTION.EQ.10)GO TO 999
      GO TO 1
```

Figure B-2 (continued)

```
C
C       THESE ARE THE FORMAT STATEMENTS FOR THE MENU
C
 199    FORMAT(1X,'                                             ',//,
        1      1X,'                                             ',//,
        2      1X,'                                             ',//)
 200    FORMAT(' -----------------------------------------------
        1----------'//,1X
        2,'SELECT A PROGRAM OPTION FROM THE CHOICES BELOW:')
 201    FORMAT(1X,'----------------------------------------------
        1------------')
 202    FORMAT(//' LIST THE INSTRUCTIONS FOR THE PROGRAM..
        1.............1')
 203    FORMAT(1X,'COUNT AN INDIVIDUAL......................
        1.............2')
 204    FORMAT(1X,'RUN A BACKGROUND COUNT.................
        1.............3')
 205    FORMAT(1X,'CHANGE THE CALIBRATION FACTORS.........
        1.............4')
 206    FORMAT(1X,'PRINT THE CURRENT CALIBRATION VALUES...
        1.............5')
 207    FORMAT(1X,'PLOT A GRAPH OF THE CURRENT DATA VALUES
        1.............6')
 208    FORMAT(1X,'RUN A SUPERVISORY COUNT................
        1.............7')
 210    FORMAT(1X,'TERMINATE THE PROGRAM..................
        1............10')
 211    FORMAT(1X,'---------------------------------------------
        1--------------')
 212    FORMAT(1X,'*')
 250    FORMAT(I4)
 999    STOP
        END
```

Figure B-3

FORTRAN Program INSTGEN

```
      SUBROUTINE INSTGEN(*)
      CHARACTER*72 MESSAGE
      PRINT 500
 500  FORMAT(1X,'                                    ',//,
     1'                                    ',//,
     2'                                    ',//)
      OPEN(UNIT=7,FILE='DUA0:[WHOLEBODY]INSTGEN.DAT',
     1STATUS='OLD',RECL=20)
      REWIND 7
 100  READ(7,300,END=200)MESSAGE
 300  FORMAT(A80)
      WRITE(6,*)MESSAGE
      GO TO 100
 200  CLOSE(7)
      RETURN 1
      END
```

The following is the data statement called by this subroutine:

GENERAL INSTRUCTIONS FOR THE RPI
WHOLEBODY COUNTING SYSTEM

THIS PROGRAM IS DESIGNED TO ALLOW THE USERS TO SELECT THOSE ROUTINES WHICH THEY WISH TO INCLUDE IN THEIR LAB OR COUNTING SESSION.

THIS PROGRAM CONTAINS SEVERAL SUBROUTINES WHICH PREPARE THE PROGRAM FOR OPERATION. RUNNING THESE IS NOT REQUIRED FOR EACH INDIVIDUAL COUNTING SESSION. HOWEVER, THEY SHOULD BE RUN AT THE BEING OF EACH LAB SESSION OR COUNTING PERIOD TO INSURE THAT THE CALIBRATION AND BACKGROUND VALUES USED REFLECT THE CURRENT SITUATION.

Figure B-3 (continued)

TO SELECT A PROGRAM OPTION SIMPLY ENTER THE NUMBER OF THE OPTION FOLLOWED BY A CARRAIGE RETURN. ALL NUMERICAL DATA THAT IS REQUESTED SHOULD BE ENTERED IN AN INTEGER FORMAT. OTHER DATA MAY BE ENTERED IN ANY CONVIENT FORMAT. FOR EXAMPLE THE DATE MAY BE ENTERED AS 10 NOVEMBER 1984/// NOVEMBER 10, 1984///11-10-84///. ONCE THE DATA IS COMPLETE ENTER A CARRAIGE RETURN TO CONTINUE THE PROGRAM.

TO START THE DETECTORS MOVING THE PERSON BEING COUNTED MUST PRESS THE BUTTON ON THE LOWER LEFT CORNER OF THE DETECTOR CARRIER.

CALIBRATION IS DONE USING A CS137 SOURCE AND ALLOWING THE DETECTORS TO REMAIN IN ONE POSITION.

A PASSWORD IS REQUIRED TO ADJUST THE CALIBRATION FACTORS FOR THE OTHER PARAMETERS OF THE PROGRAM. THESE PARAMETERS NEED ONLY BE ADJUSTED FOLLOWING A CHANGE OF CRYSTAL OR A CHANGE OF SOME OTHER MAJOR COMPONENT OF THE SYSTEM.

BACKGROUND COUNTS ARE ALSO NOT REQUIRED FOR EACH INDIVIDUAL COUNT. HOWEVER, THE TANK SHOULD REMAIN EMPTY AND ALL SOURCES SHOULD BE PUT AWAY PRIOR TO STARTING ANY BACKGROUND COUNT SO THAT LEVELS ARE NOT ARTIFICALLY HIGH.

ADDITIONAL QUESTIONS SHOULD BE ADDRESSED TO YOUR LAB ASSISTANT OR PROFESSOR RYAN.

Figure B-4

FORTRAN Program BACKGND

```
      SUBROUTINE  BACKGND(*)
      DIMENSION BKY(300),PEAK(300),IAREA2(300),E(300)
      CHARACTER*10 VARIABLE
      INTEGER*2 OPTION
      INTEGER*4 BKY,PEAK
C
C     THIS OPENS THE DATA FILE
C

      OPEN(10,FILE='DUA0:[WHOLEBODY]BKG.DAT',
      1STATUS='UNKNOWN')
C
C     THIS SECTION PRINTS THE MENU
C
 200  PRINT 100
      PRINT 101
      PRINT 102
      PRINT 103
      PRINT 104
C
C     THESE ARE THE FORMATS FOR THE MENU
C
 100  FORMAT(1X,'                                          ',/,
      1'+                                                   ',/,
      2'+                                                   ',/,
      3'    BACKGROUND CALCULATION PROGRAM              ',//)
 101  FORMAT(1X,'     ENTER YOUR OPTION AS FOLLOWS:')
 102  FORMAT(1X,'INSTRUCTIONS......................0')
 103  FORMAT(1X,'RUN BACKGROUND COUNT..............1')
 104  FORMAT(1X,'RETURN TO MAIN PROGRAM............2',/,
      1' *')
 105  FORMAT(I2)
C
C     THIS SECTION READS THE OPTION AND MOVES YOU
C     TO THE CORRECT LOCATION WITHIN THE PROGRAM
C
      READ(6,105)ICHOICE
      IF(ICHOICE.EQ.0)CALL INSTBKG(*200)
      IF(ICHOICE.EQ.1)GO TO 201
      IF(ICHOICE.EQ.2)GO TO 999
      GO TO 200
```

Figure B-4 (continued)

```
 201    PRINT 202
C
C       SHORT INSTRUCTIONS
C
 202    FORMAT(1X,'                                             ',/,
        1'+                                                     ',/,
        2'+                                                     ',/,
        3' ONE PERSON SHOULD START THE DETECTORS AS THE',/,
        4' SECOND PERSON ENTERS A  "$"  ON THE TERMINAL',/,
        5' THE PROCESS TAKES APPROXIMATELY  25  MINUTES',/,
        6' ONCE THE "$" HAS BEEN ENTEREDON THE TERMINAL.',/,
        7' *')
C
C       THIS SECTION ACTUALLY MEASURES THE BACKGROUND
C
        DO 203 I=1,260
        BKY(I)=0
 203    CONTINUE
 205    READ(6,204)IK
        IF(IK.GT.260)GO TO 207
        BKY(IK)=BKY(IK)+1
        GO TO 205
 204    FORMAT(I8)
 207    DO 206 I=1,260
        WRITE(10,204)BKY(I)
 206    CONTINUE
C
C       THIS SECTION OF THE PROGRAM SEARCHES FOR THE PEAKS
C
        K=0
        DO 800 I=4,257
        IL1=(BKY(I+1)+BKY(I+2))/2
        IL2=(BKY(I+2)+BKY(I+3))/2
        IR1=(BKY(I-1)+BKY(I-2))/2
        IR2=(BKY(I-2)+BKY(I-3))/2
C
C       COMPARES CURRENT VALUE TO NEXT THREE VALUES
C
        IF(BKY(I).GT.IL1 .AND. BKY(I).GT.IL2)GO TO 801
 800    CONTINUE
        GO TO 812
```

Figure B-4 (continued)

```
C
C       COMPARES CURRENT VALUE TO THE PREVIOUS THREE VALUES
C
 801    IF(BKY(I).GT.IR1 .AND. BKY(I).GT.IR2)GO TO 802
        GO TO 800
C
C       STORES THE LOCATION OF THE PEAK
C
 802    K=K+1
        PEAK(K)=I
        GO TO 800
 812    IWIDTH=3
        DO 804 I=1,K
C
C       CORRECTS FOR THE AREA UNDER THE CURVE
C
        IX1=PEAK(I)-IWIDTH
        IX2=PEAK(I)+IWIDTH
        IAREA2(K)=0
        IUNDER=0
        IAREA=0
        DO 805 L=IX1,IX2
        IAREA=IAREA+BKY(L)
  805   CONTINUE
        IUNDER=(1+(2*IWIDTH))*((BKY(IX1)+BKY(IX2))/2)
        IAREA2(I)=IAREA-IUNDER
        IF(IAREA2(I).LT.0)IAREA2(I)=0
 804    CONTINUE
 900    PRINT 100
        PRINT 901
        OPEN(UNIT=20,FILE ='DUA0:[WHOLEBODY]CALIBRATE.DAT',
        1       STATUS = 'UNKNOWN', ORGANIZATION='INDEXED',
        2       ACCESS = 'KEYED',  RECORDTYPE = 'VARIABLE',
        3       RECL = 14, FORM = 'UNFORMATTED',
        4       KEY=(1:10:CHARACTER))
        VARIABLE='CALFAC'
        READ(20,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        CALFAC=VALUE
```

Figure B-4 (continued)

```
C
C       THIS PRINTS THE PEAK INFORMATION AFTER
C       DETERMINING IF THE PEAK IS WITHIN THE
C       THE 95% CONFIDENCE INTERVAL.
C
        PRINT 807
        DO 806 I=1,K
        E(I)=(CALFAC*PEAK(I))+20
        SIGMA=1.96*(SQRT(FLOAT(IAREA2(I))))
        IF(IAREA2(I).LT.SIGMA)GO TO 806
        IF(IAREA2(I).LE.0)GO TO 806
        WRITE(6,808) E(I),IAREA2(I)
 806    CONTINUE
 807    FORMAT(15X,'THE FOLLOWING PEAKS WERE DETECTED:',//)
 808    FORMAT(1X,'ENERGY OF PEAK ',F10.2,' KEV...........
        1          AREA UNDER THE PEAK=',I8)
 900    PRINT 901
C
C       END OF RUN MESSAGE
C
 901    FORMAT(1X,'THIS COMPLETES THE BACKGROUND COUNTING
        1         ROUTINE')
C
C       THIS TERMINATES THE PROGRAM
C
 999    CLOSE(10,STATUS='SAVE')
        CLOSE(20,STATUS='SAVE')
        RETURN 1
        END
```

Figure B-5

FORTRAN Program INSTBKG

```
      SUBROUTINE INSTBKG(*)
      CHARACTER*72 MESSAGE
      OPEN(UNIT=7,FILE='DUA0:[WHOLEBODY]INSTBKG.DAT',
     1STATUS='OLD',RECL=20)
      REWIND 7
 100  READ(7,300,END=200)MESSAGE
 300  FORMAT(A80)
      WRITE(6,*)MESSAGE
      GO TO 100
 200  CLOSE(7)
      RETURN 1
      END
```

The following are the instructions used by this subroutine:

BACKGROUND SUBROUTINE INSTRUCTIONS

THE FOLLOWING ARE THE INSTRUCTIONS FOR THE BACKGROUND SECTION OF THE PROGRAM. THIS PHASE THE SYSTEM RUNS EMPTY THE LENGTH OF THE TRACK. THE PROGRAM IS STARTED BY ENTERING A '$' ON THE TERMINAL. START THE DETECTOR WHEN INSTRUCTED BY THE MICROCOMPUTER BY PRESSING THE BUTTON UNDER THE FRONT LEFT CORNER OF THE CARRIER ASSEMBLY.

THIS PROCESS TAKES 20 MIN ONCE THE $ IS ENTERED.

CARE SHOULD BE TAKEN DURING THIS AND ALL COUNTS TO ENSURE THAT THERE ARE NO SOURCES NEAR THE TANK WHICH WOULD CAUSE ARTIFICALLY HIGH READINGS.

THE PROGRAM WILL AUTOMATICALLY RESTART ONCE THE COUNT IS COMPLETE.

Figure B-6

FORTRAN Program CAL

```
C
C       THIS SUBROUTINE CALIBRATES THE SYSTEM USING A CS137
C       SOURCE AS THE BASIS FOR CALIBRATION  OF  THE SYSTEM.
C
        SUBROUTINE CAL(*)
        DIMENSION IDATA(300),E(300)
        CHARACTER*10 VARIABLE
        CHARACTER*8  PASSWORD
        INTEGER*2 OPTION
C
C       THIS PRINTS THE MENU
C
 599    PRINT 511
        PRINT 512
        PRINT 513
        PRINT 514
        PRINT 515
C
C       THIS READS THE USER OPTION AND DIRECTS
C       YOU WITHIN THE SUBROUTINE
C
        READ(6,516)ICHOICE
        IF(ICHOICE.EQ.0)CALL INSTCAL(*599)
        IF(ICHOICE.EQ.1)GO TO 550
        IF(ICHOICE.EQ.2)GO TO 575
        IF(ICHOICE.EQ.3)RETURN 1
        GO TO 599
C
C       THIS PRINTS THE MESSAGE TO START THE COUNT
C
 550    PRINT 511
        PRINT 517
        OPEN(UNIT=8,FILE='DUA0:[WHOLEBODY]CALIBRATE.DAT',
        1       STATUS='UNKNOWN',ORGANIZATION='INDEXED',
        2       ACCESS='KEYED',RECORDTYPE='VARIABLE',
        3       FORM='UNFORMATTED',RECL=14,
        4       KEY=(1:10:CHARACTER))
```

Figure B-6 (continued)

```
C
C       THIS READS THE DATA AND CALCULATES THE VALUE
C       OF THE CALIBRATION FACTOR
C
        IMAX=0
        IPEAK=0
        DO 554 I=1,260
        IDATA(I)=0
 554    CONTINUE
 551    READ(6,521)IP
        IF(IP.EQ.300)GO TO 552
        IDATA(IP)=IDATA(IP)+1
        IF(IDATA(IP).GT.IMAX)IPEAK=IP
        IF(IDATA(IP).GT.IMAX)IMAX=IDATA(IP)
        GO TO 551
 552    CALFAC=662./(FLOAT(IPEAK))
        VARIABLE='CALFAC'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        VALUE=CALFAC
        PRINT 800,VARIABLE,VALUE
        REWRITE(8)VARIABLE,VALUE
        CLOSE(8,STATUS='SAVE')
        GO TO 599
C
C       SUPERVISORY SECTION
C
C       ONCE THE USER HAS SUCCESSFULLY ENTERED THIS
C       SECTION THE PROGRAM BRANCHES TO THE CHANGES
C       SUBROUTINE WHICH ACTUALLY ALTERS THE CALIBRATION
C       FACTORS FOR EACH ISOTOPE.
C
 575    PRINT 511
        PRINT 522
        PRINT 523
        READ(6,524)PASSWORD
        IF(PASSWORD.NE.'GARY')GO TO 598
        CALL CHANGES(*599)
```

Figure B-6 (continued)

```
C
C       THIS PRINTS THE ERROR MESSSAGE
C
 598    PRINT 511
        PRINT 525
        PRINT 526
        GO TO 599
C
C       THIS IS THE FORMAT FOR THE MENU
C
 511    FORMAT(1X,'                                      ',/,
        1'+                                           ',/,
        2'+                                           ',/,
        3'+                                          ',//)
 512    FORMAT(1X,'      CALIBRATION SUBROUTINE      ',/,
        1        '    ENTER YOUR OPTION AS FOLLOWS:',/,
        21X,'INSTRUCTIONS......................0')
 513    FORMAT(1X,'USER CALIBRATION.................1')
 514    FORMAT(1X,'SUPERVISOR CALIBRATION............2')
 515    FORMAT(1X,'RETURN TO MAIN PROGRAM............3',/,
        1' *')
 516    FORMAT(I2)
C
C       FORMATS FOR THE INSTRUCTIONS AND SUPERVISORY SECTION
C
 517    FORMAT(' PLACE THE CESIUM SOURCE UNDER THE DETECTOR
        1   ',/,' ONCE THE SOURCE IS IN PLACE ENTER A "#"
        2ON THE',/,' TERMINAL.  THIS ROUTINE TAKES
        3APPROXIMATELY   ',/,' 15 MINUTES ONCE THE # IS
        4ENTERED',/,' *')
 521    FORMAT(I4)
 522    FORMAT(1X,'ENTER THE SUPERVISORY PASSWORD')
 523    FORMAT(1X,'EEEEEEEEEEEE',/,'+HHHHHHHHHHHH',/,
        1      '+XXXXXXXXXXXX',/,'+ ',$)
 524    FORMAT(A8)
 525    FORMAT(1X,'THE PASSWORD YOU ENTERED IS INCORRECT.')
 526    FORMAT(1X,'PLEASE CHECK BEFORE ATTEMPTING TO
        1 REACCESS THIS PROGRAM')
 527    FORMAT(1X,'WHAT CALIBRATION DATA DO YOU WISH
        1 TO CHANGE?')
 800    FORMAT(1X,'THE CURRENT VALUE OF ',A10,' IS ',F10.4)
        END
```

AD-A155 842 WHOLEBODY RADIATION COUNTING SYSTEM(U) ARMY MILITARY PERSONNEL CENTER ALEXANDRIA VA G M FECHTER 01 MAY 85

UNCLASSIFIED F/G 6/18 NL

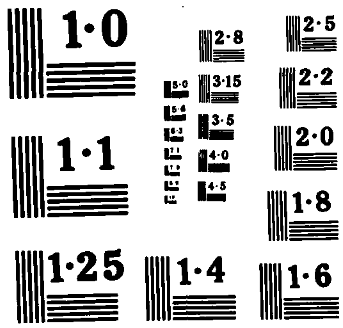

1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

Figure B-7

FORTRAN Program INSTCAL

```
      SUBROUTINE INSTCAL(*)
      CHARACTER*72 MESSAGE
      OPEN(UNIT=7,FILE='DUA0:[WHOLEBODY]INSTCAL.DAT',
     1STATUS='OLD',RECL=20)
      REWIND 7
100   READ(7,300,END=200)MESSAGE
300   FORMAT(A80)
      WRITE(6,*)MESSAGE
      GO TO 100
200   CLOSE(7)
      RETURN 1
      END
```

The following are the instructions used by this subroutine:

CALIBRATION SUBROUTINE INSTRUCTIONS

THIS SUBROUTINE IS DIVIDED INTO TWO MAJOR AREAS. THESE ARE THE USER CALIBRATION SECTION AND THE SUPERVISORY CALIBRATION.

THE USER SECTION IS DESIGNED TO BE USED AT THE BEGINNING OF EACH COUNTING SESSION TO INSURE THAT THE ENERGY CALIBRATION VALUES ARE CORRECT. ALL CALIBRATION VALUES ARE BASED ON A RELATIONSHIP WITH THE CESIUM 137 CALIBRATION VALUE AND AS SUCH THE CESIUM 137 SOURCE IS THE ONLY SOURCE REQUIRED FOR USER CALIBRATION.

THE SUPERVISORY CALIBRATION SECTION ALLOWS THE HEALTH PHYSICS STAFF TO ALTER THE INTERNAL CALIBRATION FACTORS FOLLOWING THE REPLACEMENT OF A CRYSTAL OR SIMILAR MAJOR ITEM. THIS SECTION IS NOT DESIGNED FOR GENERAL USE AND REQUIRES A SPECIAL PASSWORD TO GAIN ACCESS TO THIS SECTION.

Figure B-8

FORTRAN Program CHANGES

```
      SUBROUTINE CHANGES(*)
      CHARACTER*10 VARIABLE,CALFAC
      REAL*4 VALUE
C
C     THIS OPENS THE INDEXED FILE THAT CONTAINS ALL
C     OF THE CALIBRATION DATA
C
      OPEN(UNIT=8,FILE='DUA0:[WHOLEBODY]CALIBRATE.DAT',
      1       STATUS='UNKNOWN',ORGANIZATION='INDEXED',
      2       ACCESS='KEYED',RECORDTYPE='VARIABLE',
      3       FORM='UNFORMATTED',RECL=14,
      4       KEY=(1:10:CHARACTER))
C
C     THIS PRINTS THE MENU
C
 22   PRINT 99
C
C     FORMAT FOR THE MENU
C
 99   FORMAT('                                              ',/,
      1'+                                               ',/,
      2'+                                               ',/,
      31X,'          CALIBRATION SUBROUTINE          ',//,
      41X,'       SELECT FROM THE OPTIONS BELOW:'      ,//,
      51X,'ENTER A NEW CALIBRATION VALUE...........1',/,
      61X,'PRINT THE CURRENT CALIBRATION VALUE.....2',/,
      71X,'CHANGE THE EXISTING CALIBRATION VALUE...3',/,
      81X,'RETURN TO THE PROGRAM...................4',/,
      9' *')
C
C     THIS READS THE OPTION AND MOVES YOU WITHIN
C     THE SUBROUTINE THE APPRORIATE SECTION
C
      READ(6,12)IOPTION
      IF(IOPTION.EQ.1)GO TO 19
      IF(IOPTION.EQ.2)GO TO 20
      IF(IOPTION.EQ.3)GO TO 21
      IF(IOPTION.EQ.4)GO TO 24
      GO TO  22
```

Figure B-8 (continued)

```
C
C       THIS SECTION INITIALIZES THE CALIBRATION VALUES
C
 19     PRINT 100
        READ(6,10)VARIABLE
        PRINT 101
        READ(6,11)VALUE
        WRITE(UNIT=8)VARIABLE,VALUE
        GO TO 22
 100    FORMAT(1X,'ENTER THE VARIABLE NAME',/' *')
 101    FORMAT(1X,'ENTER THE VALUE',/' *')
C
C       THIS SECTION PRINTS CURRENT VALUES OF
C       THE CALIBRATION FACTORS
C
 20     PRINT 2
        READ(6,10)VARIABLE
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 1,VARIABLE,VALUE
        GO TO 22
 1      FORMAT(1X,'                                         ',/,
        1'+                                         ',/,
        2' THE CALIBRATION FACTOR IS DEFINED AS ',A10,/,
        3' AND  HAS A CURRENT VALUE OF ',F10.4,//)
 2      FORMAT(1X,'                                   ',/,
        1'+                                   ',/,
        2'+ENTER THE NAME OF THE VARIABLE YOU WISH
        3 PRINTED',/,' *')
C
C       THIS SECTION CHANGES THE VALUES OF
C       CALIBRATION FACTORS
C
 21     PRINT 200
        READ(6,10)VARIABLE
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 201,VARIABLE,VALUE
        PRINT 202
        READ(6,11)VALUE
        REWRITE(8)VARIABLE,VALUE
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 1,VARIABLE,VALUE
        GO TO 22
```

Figure B-8 (continued)

```
 200    FORMAT(1X,'                                                  ',/,
        1'+                                                          ',/,
        2'+ENTER THE VARIABLE YOU WISH TO CHANGE',//,'  * ')
 201    FORMAT(1X,'                                                  ',/,
        1'+                                                          ',/,
        2'+THE CURRENT VALUE OF ',A10,' IS ',F10.4)
 202    FORMAT(1X,'                                         ',/,
        1'+                                                 ',/,
        2'+ENTER THE NEW VALUE',/' *')
C
C       THESE ARE THE READ FORMATS
C
 10     FORMAT(A10)
 11     FORMAT(F10.4)
 12     FORMAT(I2)
C
C       THIS TERMINATES THE PROGRAM
C
 24     CLOSE(8,STATUS='SAVE')
        RETURN 1
        END
```

Figure B-9

FORTRAN Program CALPRINT

```
C
C       THIS SUBROUTINE PRINTS THE CURRENT VALUES
C       OF THE CALIBRATION FACTORS
C
        SUBROUTINE CALPRINT(*)
        CHARACTER*10 VARIABLE
        INTEGER*2 OPTION
        OPEN(UNIT=8,FILE='DUA0:[WHOLEBODY]CALIBRATE.DAT',
        1       STATUS='OLD',ORGANIZATION='INDEXED',
        2       ACCESS='KEYED',RECORDTYPE='VARIABLE',
        3       FORM='UNFORMATTED',RECL=14,
        4       KEY=(1:10:CHARACTER))
C
C       THIS PRINTS THE MENU
C
 1      PRINT 99
        PRINT 100
C
C       THIS READS YOUR OPTION AND MOVES
C       YOU WITHIN THE PROGRAM
C
        READ(6,101) OPTION
        IF(OPTION.EQ.1)GO TO 2
        IF(OPTION.EQ.2)GO TO 3
        IF(OPTION.EQ.3)GO TO 999
        GO TO 1
C
C       THIS SECTION PRINTS THE CALIBRATION FACTOR
C       FOR THE CS137 SOURCE
C
 2      VARIABLE='CALFAC'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 99
        PRINT 102,VALUE
        GO TO 1
```

Figure B-9 (continued)

```
.
C
C    THIS SECTION PRINTS THE ENTIRE LIST
C       OF CALIBRATION VALUES
C
 3      PRINT 99
        VARIABLE='CALFAC'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 201,VALUE
        VARIABLE='MNCAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 202,VALUE
        VARIABLE='CSCAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 203,VALUE
        VARIABLE='OICAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 204,VALUE
        VARIABLE='CRCAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 205,VALUE
        VARIABLE='CO58CAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 206,VALUE
        VARIABLE='CO60CAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 207,VALUE
        VARIABLE='ZNCAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 208,VALUE
        VARIABLE='POTCAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 209,VALUE
        VARIABLE='NACAL'
        READ(8,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        PRINT 210,VALUE
        GO TO 1
```

Figure B-9 (continued)

```
C
C       FORMATS USED FOR THE MENU AND OUTPUT STATEMENTS
C
 99     FORMAT(1X,'                                            ',/,
        1'+                                                    ',/,
        2'+                                                    ',/)
 100    FORMAT(1X,'SELECT THE OPTION FROM THE CHOICES
        1BELOW:',//,1X,'PRINT THE SOURCE CALIBRATION
        2VALUE ONLY...........1',/,1X,'PRINT THE
        3COMPLETE LIST OF CALIBRATION FACTORS....2',/,
        41X,'RETURN TO THE MAIN PROGRAM.............
        5..........3',/,1X,'*')
 101    FORMAT(I4)
 102    FORMAT(1X,'THE ENERGY CALIBRATION FACTOR
        1CURRENTLY BEING ',/,1X,'USED BY THIS PROGRAM
        2IS ',F10.4,//)
 201    FORMAT(1X,'THE FOLLOWING ARE THE VALUES OF THE
        1CALIBRATION',/,1X,'FACTORS CURRENTLY BEING USED
        3BY THIS PROGRAM',//,1X,'THE ENERGY CALIBRATION
        4FACTOR EQUALS ',F10.4,//)
 202    FORMAT(1X,'THE CURRENT VALUES FOR THE FACTORS USED
        1FOR',/,1X,'EACH RADIONUCLIDE ARE AS FOLLOWS:',//,
        21X,'MANGENESE 54  CALIBRATION FACTOR   ',F10.4)
 203    FORMAT(1X,'CESIUM 137    CALIBRATION FACTOR   ',
        1F10.4)
 204    FORMAT(1X,'IODINE 131    CALIBRATION FACTOR   ',
        1F10.4)
 205    FORMAT(1X,'CHROMIUM 51   CALIBRATION FACTOR   ',
        1F10.4)
 206    FORMAT(1X,'COBALT 58     CALIBRATION FACTOR   ',
        1F10.4)
 207    FORMAT(1X,'COBALT 60     CALIBRATION FACTOR   ',
        1F10.4)
 208    FORMAT(1X,'ZINC 65       CALIBRATION FACTOR   ',
        1F10.4)
 209    FORMAT(1X,'POTASSIUM 40  CALIBRATION FACTOR   ',
        1F10.4)
 210    FORMAT(1X,'SODIUM 22     CALIBRATION FACTOR   ',
        1F10.4,//)
 999    CLOSE(8)
        RETURN 1
        END
```

Figure B-10

FORTRAN Program COUNT

```
      SUBROUTINE COUNT(*)
      CHARACTER*10 VARIABLE
      CHARACTER*40 NAME
      CHARACTER*20 DATE
      CHARACTER*1 SEX
      INTEGER*2 OPTION
      INTEGER*4 DATA,HEIGHT,WEIGHT,AGE,YMAX,Y,BKY,X,E,PEAK
      DIMENSION DATA(300),BKY(300),Y(300),PEAK(300),IAREA2
     1          (300),E(300)
C
C     THIS PRINTS MENU
C
 99   PRINT 100
      PRINT 11
      PRINT 12
      PRINT 13
      PRINT 14
C
C     THIS IS THE FORMAT FOR THE MENU
C
 100  FORMAT(1X,'                                        ',/,
     1'+                                               ',/,
     2'+                                             ',/,/)
 11   FORMAT(1X,'         COUNTING SUBROUTINE          ',/,
     1'   ENTER THE OPTION FROM LIST BELOW:          ')
 12   FORMAT(1X,'INSTRUCTIONS.........................0')
 13   FORMAT(1X,'COUNT AN INDIVIDAL...................1')
 14   FORMAT(1X,'RETURN TO THE PROGRAM................2',
     1/,' *')
C
C     READ OPTION
C
      READ(6,15)OPTION
      IF(OPTION.EQ.0)CALL INSTCNT(*99)
      IF(OPTION.EQ.1)GO TO 600
      IF(OPTION.EQ.2)GO TO 999
      GO TO 99
 15   FORMAT(I2)
```

Figure B-10 (continued)

```
C
C       THE MAIN COUNTING PROGRAM
C
 600    OPEN(UNIT=10,FILE='DUA0:[WHOLEBODY]CALIBRATE.DAT',
        1       STATUS='UNKNOWN', ORGANIZATION= 'INDEXED',
        2       ACCESS = 'KEYED', RECORDTYPE = 'VARIABLE',
        3       FORM= 'UNFORMATTED', KEY= (1:10:CHARACTER)
        4       RECL = 14)
        OPEN(UNIT=12,FILE='DUA0:[WHOLEBODY]DATA.DAT',
        1       STATUS='UNKNOWN')
        VARIABLE='CALFAC'
        READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        CALFAC=VALUE
C
C       THIS SECTIONS COLLECTS THE PERSONAL
C       DATA ON  THE  PERSON  BEING COUNTED
C
        INDEX=0
        PRINT 100
        PRINT 101
        READ(6,102)NAME
        PRINT 100
        PRINT 103
        READ(6,104)ISSN
        PRINT 100
        PRINT 105
        READ(6,106)DATE
        PRINT 100
        PRINT 114
        READ(6,115)HEIGHT
        PRINT 100
        PRINT 116
        READ(6,117)WEIGHT
        PRINT 100
        PRINT 118
        READ(6,119)SEX
        PRINT 100
        PRINT 120
        READ(6,121)AGE
```

Figure B-10 (continued)

```
C
C       INSTRUCTION AND DATA FORMAT STATEMENTS
C
 101    FORMAT(/1X,'ENTER THE NAME OF THE PERSON BEING
        1COUNTED',/' *')
 102    FORMAT(A40)
 103    FORMAT(/1X,'ENTER THE SOCIAL SECURITY NUMBER',/,
        1' *')
 104    FORMAT(I9)
 105    FORMAT(/1X,'ENTER THE DATE',/' *')
 106    FORMAT(A20)
 114    FORMAT(/1X,'ENTER YOUR HEIGHT IN INCHES',/' *')
 115    FORMAT(I2)
 116    FORMAT(/1X,'ENTER YOUR WEIGHT IN POUNDS',/' *')
 117    FORMAT(I3)
 118    FORMAT(/1X,'ENTER SEX AS M OR F',/' *')
 119    FORMAT(A1)
 120    FORMAT(/1X,'ENTER YOUR AGE IN YEARS',/' *')
 121    FORMAT(I2)
        PRINT 100
        PRINT 107

C       SHORT INSTRUCTIONS
C
 107    FORMAT(/1X,' HAVE THE  PERSON  BEING  COUNTED LIE',
        1' DOWN IN THE  TANK  WITH  THEIR',/,' FEET UNDER',
        2' THE  DETECTORS.     THE  PERSON  BEING COUNTED',
        3' SHOULD  START   THE   DETECTORS AS THE  SECOND',
        4' PERSON   ENTERS   A   "$"  ON  THE  TERMINAL',/,'
        5' ONCE  THE  DETECTOR  STOPS  THE  PERSON  BEING',
        6' COUNTED SHOULD RESTART  IT  WITH  THE  BUTTON ',
        7' IN  THE  FRONT  RIGHT  CORNER  OF THE CARRIER.',
        8' THE COUNT WILL TAKE  APPROXIMATELY  25 MINUTES',
        9' ONCE THE "$" IS ENTERED ON THE TERMINAL.',/,' *)
 112    FORMAT(F10.2)
C
C       THIS SECTION READS THE DATA AND CALCULATES YMAX
C
        YMAX=0
        DO 200 I=1,260
        DATA(I)=0
 200    CONTINUE
```

Figure B-10 (continued)

```
 210    READ(6,201)X
        IF(X.GT.260)GO TO 220
        IF(X.EQ.0)GO TO 210
        DATA(X)=DATA(X)+1
        IF(DATA(X).GT.YMAX)YMAX=DATA(X)
        GO TO 210
 201    FORMAT(I8)
C
C       THIS SECTION SUBTRACTS BACKGROUND AND STORES
C       THE DATA IN A FILE CALLED BKG.DAT
C
 220    OPEN(UNIT=11,FILE='DUA0:[WHOLEBODY]BKG.DAT',
        1STATUS='UNKNOWN')
        WRITE(12,201)YMAX
        DO 202  I=1,260
        READ(11,201)BKY(I)
        Y(I)=DATA(I)-BKY(I)
        WRITE(12,201)Y(I)
 202    CONTINUE
        CLOSE(11,STATUS='SAVE')
        CLOSE(11,STATUS='SAVE')
C
C       THIS SECTION OF THE PROGRAM SEARCHES FOR THE PEAKS
C
        K=0
        DO 800 I=4,257
        IL1=(Y(I+1)+Y(I+2))/2
        IL2=(Y(I+2)+Y(I+3))/2
        IR1=(Y(I-1)+Y(I-2))/2
        IR2=(Y(I-2)+Y(I-3))/2
C
C       COMPARES CURRENT VALUE TO NEXT THREE VALUES
C
        IF(Y(I).GT.IL1 .AND. Y(I).GT.IL2)GO TO 801
 800    CONTINUE
        GO TO 812
C
C       COMPARES CURRENT VALUE TO THE PREVIOUS THREE VALUES
C
 801    IF(Y(I).GT.IR1 .AND. Y(I).GT.IR2)GO TO 802
        GO TO 800
```

Figure B-10 (continued)

```
C
C       STORES THE LOCATION OF THE PEAK
C
 802    K=K+1
        PEAK(K)=I
        GO TO 800
 812    IWIDTH=3
        DO 804 I=1,K
C
C       CORRECTS FOR THE AREA UNDER THE CURVE
C
        IX1=PEAK(I)-IWIDTH
        IX2=PEAK(I)+IWIDTH
        IAREA2(K)=0
        IUNDER=0
        IAREA=0
        DO 805 L=IX1,IX2
        IAREA=IAREA+Y(L)
  805   CONTINUE
        IUNDER=(1+(2*IWIDTH))*((Y(IX1)+Y(IX2))/2)
        IAREA2(I)=IAREA-IUNDER
        IF(IAREA2(I).LT.0)IAREA2(I)=0
 804    CONTINUE
C
C       THIS INITIALIZES ALL CONCENTRATIONS TO ZERO
C
        OI131=0.
        CS137=0.
        CR51=0.
        CO58=0.
        MN54=0.
        ZN65=0.
        CO60=0.
        NA22=0.
        POT40=0.
C
C       THIS SECTION CALCULATES THE NUMBER OF PULSES
C       WITHIN GIVEN ENERGY BANDS TO DETERMINE
C       CONCENTRATIONS OF SPECIFIC RADIONUCLIDES
C
 151    DO 152 I=1,K
        E(I)=(PEAK(I)*CALFAC)+20.0
        IF(E(I) .GT. 335. .AND. E(I) .LT. 370.)OI131=IAREA2(I)
```

Figure B-13 (continued)

```
C
C       STORES THE LOCATION OF THE PEAK
C
 802    K=K+1
        PEAK(K)=I
        GO TO 800
 812    IWIDTH=3
        DO 804 I=1,K
C
C       CORRECTS FOR THE AREA UNDER THE CURVE
C
        IX1=PEAK(I)-IWIDTH
        IX2=PEAK(I)+IWIDTH
        IAREA2(K)=0
        IUNDER=0
        IAREA=0
        DO 805 L=IX1,IX2
        IAREA=IAREA+Y(L)
  805   CONTINUE
        IUNDER=(1+(2*IWIDTH))*((Y(IX1)+Y(IX2))/2)
        IAREA2(I)=IAREA-IUNDER
        IF(IAREA2(I).LT.0)IAREA2(I)=0
 804    CONTINUE
C
C       THIS INITIALIZES ALL CONCENTRATIONS TO ZERO
C
        OI131=0.
        CS137=0.
        CR51=0.
        CO58=0.
        MN54=0.
        ZN65=0.
        CO60=0.
        NA22=0.
        POT40=0.
C
C       THIS SECTION CALCULATES THE NUMBER OF PULSES
C       WITHIN GIVEN ENERGY BANDS TO DETERMINE
C       CONCENTRATIONS OF SPECIFIC RADIONUCLIDES
C
 151    DO 152 I=1,K
        E(I)=(PEAK(I)*CALFAC)+20.0
        IF(E(I).GT.335..AND.E(I).LT.370.)OI131=IAREA2(I)
```

Figure B-13 (continued)

```
 210    READ(6,201)X
        IF(X.GT.260)GO TO 220
        IF(X.EQ.0)GO TO 210
        DATA(X)=DATA(X)+1
        IF(DATA(X).GT.YMAX)YMAX=DATA(X)
        GO TO 210
 201    FORMAT(I8)
C
C       THIS SECTION SUBTRACTS BACKGROUND AND STORES
C       THE DATA IN A FILE CALLED BKG.DAT
C
 220    OPEN(UNIT=11,FILE='DUA0:[WHOLEBODY]BKG.DAT',
        1STATUS='UNKNOWN')
        WRITE(12,201)YMAX
        DO 202  I=1,260
        READ(11,201)BKY(I)
        Y(I)=DATA(I)-BKY(I)
        IF(Y(I).LT.0)Y(I)=0
        WRITE(12,201)Y(I)
 202    CONTINUE
        CLOSE(11,STATUS='SAVE')
C
C       THIS SECTION OF THE PROGRAM SEARCHES FOR THE PEAKS
C
        K=0
        DO 800 I=4,257
        IL1=(Y(I+1)+Y(I+2))/2
        IL2=(Y(I+2)+Y(I+3))/2
        IR1=(Y(I-1)+Y(I-2))/2
        IR2=(Y(I-2)+Y(I-3))/2
C
C       COMPARES CURRENT VALUE TO NEXT THREE VALUES
C
        IF(Y(I).GT.IL1 .AND. Y(I).GT.IL2)GO TO 801
 800    CONTINUE
        GO TO 812
C
C       COMPARES CURRENT VALUE TO THE PREVIOUS THREE VALUES
C
 801    IF(Y(I).GT.IR1 .AND. Y(I).GT.IR2)GO TO 802
        GO TO 800
```

Figure B-13 (continued)

```
C
C       INSTRUCTION AND DATA FORMAT STATEMENTS
C
 101    FORMAT(5X,'ENTER THE NAME OF THE PERSON BEING
        1COUNTED',/' *')
 102    FORMAT(A40)
 103    FORMAT(5X,'ENTER THE SOCIAL SECURITY NUMBER',/,
        1' *')
 104    FORMAT(I9)
 105    FORMAT(5X,'ENTER THE DATE',/' *')
 106    FORMAT(A20)
 114    FORMAT(5X,'ENTER YOUR HEIGHT IN INCHES',/' *')
 115    FORMAT(I2)
 116    FORMAT(5X,'ENTER YOUR WEIGHT IN POUNDS',/' *')
 117    FORMAT(I3)
 118    FORMAT(5X,'ENTER SEX AS M OR F',/' *')
 119    FORMAT(A1)
 120    FORMAT(5X,'ENTER YOUR AGE IN YEARS',/' *')
 121    FORMAT(I2)
 500    FORMAT(5X,'ENTER THE DESIRED VALUE OF SIGMA',/' *')
 501    FORMAT(F10.5)
        PRINT 107
C
C       SHORT INSTRUCTIONS
C
 107    FORMAT(/1X,' HAVE THE  PERSON  BEING  COUNTED LIE',
        1' DOWN IN THE  TANK  WITH  THEIR',/,' FEET UNDER',
        2' THE  DETECTORS.     THE  PERSON  BEING COUNTED',
        3' SHOULD  START   THE   DETECTORS AS THE  SECOND',
        4' PERSON   ENTERS   A  "$"  ON  THE  TERMINAL',/,'
        5' ONCE  THE  DETECTOR  STOPS  THE  PERSON  BEING',
        6' COUNTED SHOULD RESTART  IT  WITH  THE  BUTTON ',
        7' IN  THE  FRONT  RIGHT  CORNER  OF THE CARRIER.',
        8' THE COUNT WILL TAKE  APPROXIMATELY  25 MINUTES',
        9' ONCE THE "$" IS ENTERED ON THE TERMINAL.',/,' *)
 112    FORMAT(F10.2)
C
C       THIS SECTION READS THE DATA AND CALCULATES YMAX
C
        YMAX=0
        DO 200 I=1,260
        DATA(I)=0
 200    CONTINUE
```

Figure B-13 (continued)

```
C
C  THE MAIN COUNTING PROGRAM
C
 600    OPEN(UNIT=10,FILE='DUA0:[WHOLEBODY]CALIBRATE.DAT',
        1       STATUS='UNKNOWN',ORGANIZATION='INDEXED',
        2       ACCESS='KEYED',RECORDTYPE='VARIABLE',RECL=14,
        3       FORM='UNFORMATTED',KEY=(1:10:CHARACTER))
        OPEN(UNIT=12,FILE='DUA0:[WHOLEBODY]DATA.DAT',
        1STATUS='UNKNOWN')
        VARIABLE='CALFAC'
        READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
        CALFAC=VALUE
C
C       THIS COLLECTS THE PERSONAL DATA ON THE PERSON
C       BEING COUNTED
C
        INDEX=0
        PRINT 100
        PRINT 101
        READ(6,102)NAME
        PRINT 100
        PRINT 103
        READ(6,104)ISSN
        PRINT 100
        PRINT 105
        READ(6,106)DATE
        PRINT 100
        PRINT 114
        READ(6,115)HEIGHT
        PRINT 100
        PRINT 116
        READ(6,117)WEIGHT
        PRINT 100
        PRINT 118
        READ(6,119)SEX
        PRINT 100
        PRINT 120
        READ(6,121)AGE
        PRINT 500
        READ(6,501)SIGMA
```

Figure B-13

```
      SUBROUTINE SUPCOUNT(*)
      CHARACTER*10 VARIABLE
      CHARACTER*40 NAME
      CHARACTER*20 DATE
      CHARACTER*1 SEX
      INTEGER*2 OPTION
      INTEGER*4 DATA,HEIGHT,WEIGHT,AGE,YMAX,Y,BKY,X,PEAK
      DIMENSION DATA(300),BKY(300),Y(300),PEAK(300),
     1          IAREA2(300),E(300)
C
C  THIS PRINTS MENU
C
 99   PRINT 100
      PRINT 11
      PRINT 12
      PRINT 13
      PRINT 14
C
C  THIS IS THE FORMAT FOR THE MENU
C
 100  FORMAT(1X,'                                  ',/,
     1'+                                       ',/,
     2'+                                       ',/,
     3'+                                       ',/,
     4'+                                       ')
 11   FORMAT(15X,'        COUNTING SUBROUTINE        ',/
     1,15X,'  ENTER THE OPTION FROM LIST BELOW:')
 12   FORMAT(15X,'INSTRUCTIONS.........................0')
 13   FORMAT(15X,'COUNT AN INDIVIDAL...................1')
 14   FORMAT(15X,'RETURN TO THE PROGRAM................2',/,
     1' *')
C
C  READ OPTION
C
      READ(6,15)OPTION
      IF(OPTION.EQ.0)CALL INSTCNT(*99)
      IF(OPTION.EQ.1)GO TO 600
      IF(OPTION.EQ.2)GO TO 999
      GO TO 99
 15   FORMAT(I2)
```

Figure B-12 (continued)

```
C
C       THIS FILLS THE GRAPH
C
        DO 50 I=1,260
        J=Y(I)+10
        TEMP=PT(I)
        IF(Y(I).GT.0)TEMP(J:J)='#'
        PT(I)=TEMP
 50     CONTINUE
C
C       THIS PRINTS THE RESULTS
C
        PRINT 99
        WRITE(10,100),YT1
        WRITE(10,101),YMAX
        WRITE(10,100),YT2
        WRITE(10,100),YT3
        WRITE(10,100),YT4
        WRITE(10,100),LINE
        DO 202  I=1,260
        WRITE(10,100),PT(I)
 202    CONTINUE
C
C       THESE ARE THE FORMAT STATEMENTS USED
C       TO PRINT THE GRAPH
C
 99     FORMAT(1X,'                                        ',//,
        11X,'                                          ',//,
        21X,'YOUR PLOT IS AVAILABLE IN THE MAIN
        3COMPUTER ROOM.',//)
 100    FORMAT(1X,A135)
 101    FORMAT(1X,'                    Y MAXIMUM  = ',I8)
 110    FORMAT(I8)
 190    FORMAT(1X,I4)
 191    FORMAT(1X,F10.3)
        CLOSE(10,STATUS='PRINT')
        CLOSE(11,STATUS='SAVE')
        RETURN 1
        END
```

Figure B-12 (continued)

```
      IF(ID.EQ.200)TEMP(7:10)='200+'
      IF(ID.EQ.210)TEMP(7:10)='210+'
      IF(ID.EQ.220)TEMP(7:10)='220+'
      IF(ID.EQ.230)TEMP(7:10)='230+'
      IF(ID.EQ.240)TEMP(7:10)='240+'
      IF(ID.EQ.250)TEMP(7:10)='250+'
      IF(ID.EQ.260)TEMP(7:10)='260+'
      PT(ID)=TEMP
 30   CONTINUE
      K=1
      XT='GAMMA ENERGY EXPRESSED AS CHANNEL NUMBERS'
      DO 201 I=40,81
      TEMP=PT(I)
      TEMP(3:3)=XT(K:K)
      K=K+1
      PT(I)=TEMP
 201  CONTINUE
C
C     THIS INITIALIZES THE SYSTEM
C
      YMAX=0
      DO 500 I=1,260
      CHANNEL(I)=0
 500  CONTINUE
C
C     THIS READS THE DATA VALUES
C
      READ(11,110)YMAX
      DO 25 I=1,260
      READ(11,110)IVAL
      CHANNEL(I)=IVAL
 25   CONTINUE
C
C     THIS CALCULATES THE VALUE AS A PERCENTAGE OF
C     THE MAXIMUM Y VALUE
C
 300  DO 501 I=1,260
      Y(I)=(CHANNEL(I)/YMAX)*100
 501  CONTINUE
```

Figure B-12 (continued)

```
      YT3=BLANK
      YT3(4:4)=' '
      YT3(21:21)='1'
      YT3(31:31)='2'
      YT3(41:41)='3'
      YT3(51:51)='4'
      YT3(61:61)='5'
      YT3(71:71)='6'
      YT3(81:81)='7'
      YT3(91:91)='8'
      YT3(101:101)='9'
      YT3(111:111)='0'
      YT4=BLANK
      YT4(4:4)=' '
      DO 200 I=1,10
      ID=11+(I*10)
      YT4(ID:ID)='0'
      LINE(ID:ID)='+'
 200  CONTINUE
C
C     THIS ESTABLISHES THE X HEADINGS
C
      DO 30 I=1,26
      TEMP=BLANK
      ID=10*I
      IF(ID.EQ.10)TEMP(8:10)='10+'
      IF(ID.EQ.20)TEMP(8:10)='20+'
      IF(ID.EQ.30)TEMP(8:10)='30+'
      IF(ID.EQ.40)TEMP(8:10)='40+'
      IF(ID.EQ.50)TEMP(8:10)='50+'
      IF(ID.EQ.60)TEMP(8:10)='60+'
      IF(ID.EQ.70)TEMP(8:10)='70+'
      IF(ID.EQ.80)TEMP(8:10)='80+'
      IF(ID.EQ.90)TEMP(8:10)='90+'
      IF(ID.EQ.100)TEMP(7:10)='100+'
      IF(ID.EQ.110)TEMP(7:10)='110+'
      IF(ID.EQ.120)TEMP(7:10)='120+'
      IF(ID.EQ.130)TEMP(7:10)='130+'
      IF(ID.EQ.140)TEMP(7:10)='140+'
      IF(ID.EQ.150)TEMP(7:10)='150+'
      IF(ID.EQ.160)TEMP(7:10)='160+'
      IF(ID.EQ.170)TEMP(7:10)='170+'
      IF(ID.EQ.180)TEMP(7:10)='180+'
      IF(ID.EQ.190)TEMP(7:10)='190+'
```

Figure B-12

FORTRAN Program PLOT

```
      SUBROUTINE PLOT(*)
      DIMENSION PT(300),Y(300),CHANNEL(300)
      INTEGER*2 ID
      INTEGER*4 X,YMAX
      CHARACTER*135 PT,LINE,BLANK,TEMP,YT1,YT2,YT3,YT4,XT
C
C     THIS OPENS THE REQUIRED FILES
C
      OPEN(10,FILE='DUA0:[WHOLEBODY]PLOTING.DAT',
     1        STATUS='UNKNOWN',RECL=140)
      OPEN(11,FILE='DUA0:[WHOLEBODY]DATA.DAT',
     1        STATUS='UNKNOWN')
C
C     THIS ESTABLISHES THE BLANK AND DOTTED LINES
C
      BLANK(1:11)='          !'
      DO 11 I=11,135
      BLANK(I:I)=' '
 11   CONTINUE
      LINE(1:10)='          '
      DO 12 I=11,111
      LINE(I:I)='-'
 12   CONTINUE
C
C     THIS CONSTRUCTS THE Y HEADINGS
C
      DO 13 I=1,21
      YT1(I:I)=' '
 13   CONTINUE
      YT1(20:75)='NUMBER OF COUNTS OBSERVED AS A
     1        PERCENTAGE OF Y MAX'
      DO 20 I=1,260
      PT(I)=BLANK
 20   CONTINUE
      YT2=BLANK
      YT2(4:4)=' '
      YT2(111:111)='1'
```

Figure B-11

FORTRAN Program INSTCNT

```
      SUBROUTINE INSTCNT(*)
      CHARACTER*72 MESSAGE
      OPEN(UNIT=7,FILE='DUA0:[WHOLEBODY]INSTCNT.DAT',
     1STATUS='OLD',RECL=20)
      REWIND 7
100   READ(7,300,END=200)MESSAGE
300   FORMAT(A80)
      WRITE(6,*)MESSAGE
      GO TO 100
200   CLOSE(7)
      RETURN 1
      END
```

The following are the instructions for this subroutine:

COUNTING SUBROUTINE INSTRUCTIONS

IN THE INITIAL PHASE OF THE COUNTING ROUTINE YOU WILL BE ASKED TO ENTER YOUR NAME, SOCIAL SECURITY NUMBER, THE DATE AND OTHER INFORMATION.

THIS INFORMATION MAY BE ENTERED IN ANY MANNER DESIRED. THE PROGRAM WILL ACCEPT A MIX OF ALPHABETIC AND NUMERIC CHARACTERS.

THIS INFORMATION SHOULD BE AS ACCURATE AS POSSIBLE AS THE PRINTOUT WILL BE YOUR ONLY RECORD OF THE COUNTING SESSION AND IT WILL DUPLICATE THIS INFORMATION.

Figure B-10 (continued)

```
 160    FORMAT(1X,'
        1 RENSSELAER POLYTECHNIC INSTITUTE ')
 161    FORMAT(1X,'
        1WHOLEBODY RADIATION COUNTING SYSTEM')
 162    FORMAT(1X,'-------------------------------------
        1-------------')
 163    FORMAT(//1X,A20,'WAS COUNTED ON',2X,A20)
 176    FORMAT(/1X,'THE FOLLOWING INFORMATION IS PROVIDED:')
 164    FORMAT(1X,'SOCIAL SECURITY ACCOUNT NUMBER   ',2X,I9)
 172    FORMAT(1X,'HEIGHT..................',I4,2X,'INCHES')
 173    FORMAT(1X,'WEIGHT..................',I4,2X,'POUNDS')
 174    FORMAT(1X,I2,'  YEARS OLD                  FEMALE',//)
 175    FORMAT(1X,I2,'  YEARS OLD                    MALE',//)
 177    FORMAT(1X,'-------------------------------------
        1-------------')
 178    FORMAT(1X,'QUANTITIES OF RADIOISOTOPES PRESENT:'
        1,//)
 165    FORMAT(1X,'QUANTITY OF CESIUM 137 PRESENT  '
        1,F12.6,'  NANOCURIES')
 166    FORMAT(1X,'QUANTITY OF IODINE 131 PRESENT  '
        1,F12.6,'  NANOCURIES')
 167    FORMAT(1X,'QUANTITY OF CHROMIUM 51 PRESENT '
        1,F12.6,'  NANOCURIES')
 168    FORMAT(1X,'QUANTITY OF COBALT 58 PRESENT   '
        1,F12.6,'  NANOCURIES')
 169    FORMAT(1X,'QUANTITY OF COBALT 60 PRESENT   '
        1,F12.6,'  NANOCURIES')
 170    FORMAT(1X,'QUANTITY OF ZINC 65 PRESENT     '
        1,F12.6,'  NANOCURIES')
 180    FORMAT(1X,'QUANTITY OF POTASSIUM 40 PRESENT'
        1,F12.6,'  NANOCURIES')
 182    FORMAT(1X,'QUANTITY OF SODIUM 22 PRESENT   '
        1,F12.6,'  NANOCURIES')
 181    FORMAT(1X,'QUANTITY OF RADIATION PRESENT IS BELOW
        1 THE LOWER',/,' LIMIT OF DETECTION FOR THIS
        2SYSTEM.')
 171    FORMAT(1X,'-------------------------------------
        1-----------',//)
        GO TO 99
 999    CLOSE(10,STATUS='SAVE')
        CLOSE(12,STATUS='SAVE')
        RETURN 1
        END
```

Figure B-10 (continued)

```
C
C       THIS PRINTS THE PEAK INFORMATION AFTER
C       DETERMINING IF THE PEAK IS WITHIN THE
C       THE 95% CONFIDENCE INTERVAL.
C
        PRINT 807
        DO 806 I=1,K
        SIGMA=1.96*(SQRT(FLOAT(IAREA2(I))))
        IF(IAREA2(I).LT.SIGMA)GO TO 806
        IF(IAREA2(I).GT.0)WRITE(6,808) E(I),IAREA2(I)
 806    CONTINUE
        PRINT 177
 807    FORMAT(15X,'THE FOLLOWING PEAKS WERE DETECTED:',//)
 808    FORMAT(1X,'ENERGY OF PEAK ',F10.2,' KEV.....
        1....AREA UNDER THE PEAK=',I8)
C
C       THIS PRINTS ONLY THOSE CONCENTRATIONS GREATER
C       THAN THE LOWER LIMIT FOR THE DETECTOR SYSTEM.
C
        IF(CICS.GT.0.)INDEX=INDEX+1
        IF(CICS.GT.0.)PRINT 165,CICS
        IF(CIOI.GT.0.)INDEX=INDEX+1
        IF(CIOI.GT.0.)PRINT 166,CIOI
        IF(CICR.GT.0.)INDEX=INDEX+1
        IF(CICR.GT.0.)PRINT 167,CICR
        IF(CIC58.GT.0.)INDEX=INDEX+1
        IF(CIC58.GT.0.)PRINT 168,CIC58
        IF(CIC60.GT.0.)INDEX=INDEX+1
        IF(CIC60.GT.0.)PRINT 169,CIC60
        IF(CIZN.GT.0.)INDEX=INDEX+1
        IF(CIZN.GT.0.)PRINT 170,CIZN
        IF(CIK40.GT.0.)INDEX=INDEX+1
        IF(CIK40.GT.0.)PRINT 180,CIK40
        IF(CNA22.GT.0.)INDEX=INDEX+1
        IF(CNA22.GT.0.)PRINT 182,CNA22
        IF(INDEX.EQ.0)PRINT 181
        PRINT 171
C
C       OUTPUT FORMAT STATEMENTS
C
 159    FORMAT(///,'-----------------------------------
        1--------------')
```

Figure B-10 (continued)

```
      VARIABLE='CRCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CICR=CR51*VALUE
      VARIABLE='CO58CAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIC58=CO58*VALUE
      VARIABLE='CO60CAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIC60=CO60*VALUE
      VARIABLE='ZNCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIZN=ZN65*VALUE
      VARIABLE='POTCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIK40=POT40*VALUE
      VARIABLE='NACAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CNA22=NA22*VALUE
C
C     THIS SECTION PRINTS THE RESULTS HEADING
C
      PRINT 100
      PRINT 159
      PRINT 160
      PRINT 161
      PRINT 162
      PRINT 176
      PRINT 163,NAME,DATE
      PRINT 164,ISSN
      PRINT 172,HEIGHT
      PRINT 173,WEIGHT
      IF(SEX.EQ.'F')PRINT 174,AGE
      IF(SEX.EQ.'f')PRINT 174,AGE
      IF(SEX.EQ.'M')PRINT 175,AGE
      IF(SEX.EQ.'m')PRINT 175,AGE
      PRINT 177
      PRINT 178
```

Figure B-10 (continued)

```
      IF(E(I).GT.632..AND.E(I).LT.692.)CS137=IAREA2(I)
      IF(E(I).GT.720..AND.E(I).LT.780.)CR51=IAREA2(I)
      IF(E(I).GT.781..AND.E(I).LT.825.)CO58=IAREA2(I)
      IF(E(I).GT.826..AND.E(I).LT.875.)MN54=IAREA2(I)
C     SODIUM 22 CALCULATION
      IF(E(I).GT.218..AND.E(I).LT.278.)NA22=IAREA2(I)+NA22
      IF(E(I).GT.729..AND.E(I).LT.789.)NA22=IAREA2(I)+NA22
      IF(E(I).GT.1240..AND.E(I).LT.1300.)NA22=IAREA2(I)+NA22
C     COBALT 60 CALCULATION
      IF(E(I).GT.122..AND.E(I).LT.182.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.290..AND.E(I).LT.330.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.633..AND.E(I).LT.693.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.801..AND.E(I).LT.841.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.1144..AND.E(I).LT.1204.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.1312..AND.E(I).LT.1352.)CO60=CO60+IAREA2(I)
C     ZINC 65 CALCULATION
      IF(E(I).GT.63..AND.E(I).LT.121.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.308..AND.E(I).LT.353.)ZN65=ZN65+IAREA2(I)
      IF(E(I).GT.574..AND.E(I).LT.632.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.841..AND.E(I).LT.864.)ZN65=ZN65+IAREA2(I)
      IF(E(I).GT.1085..AND.E(I).LT.1143.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.1330..AND.E(I).LT.1375.)ZN65=ZN65+IAREA2(I)
C     POTASSIUM 40 CALCULATION
      IF(E(I).GT.438..AND.E(I).LT.478.)POT40=POT40+IAREA2(I)
      IF(E(I).GT.949..AND.E(I).LT.989.)POT40=POT40+IAREA2(I)
      IF(E(I).GT.1460..AND.E(I).LT.1500.)POT40=POT40+IAREA2(I)
  152 CONTINUE
C
C     THIS READS THE CALIBRATION VALUES AND CALCULATES
C     THE CONCENTRATION FOR EACH ISOTOPE PRESENT
C     THESE ARE MEASURED IN NANOCURIES
C
      VARIABLE='MNCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIMN=MN54*VALUE
      VARIABLE='CSCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CICS=CS137*VALUE
      VARIABLE='OICAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIOI=OI131*VALUE
```

Figure B-13 (continued)

```
      IF(E(I).GT.632..AND.E(I).LT.692.)CS137=IAREA2(I)
      IF(E(I).GT.720..AND.E(I).LT.780.)CR51=IAREA2(I)
      IF(E(I).GT.781..AND.E(I).LT.825.)CO58=IAREA2(I)
      IF(E(I).GT.826..AND.E(I).LT.875.)MN54=IAREA2(I)
C     SODIUM 22 CALCULATION
      IF(E(I).GT.218..AND.E(I).LT.278.)NA22=IAREA2(I)+NA22
      IF(E(I).GT.729..AND.E(I).LT.789.)NA22=IAREA2(I)+NA22
      IF(E(I).GT.1240..AND.E(I).LT.1300.)NA22=IAREA2(I)+NA22
C     COBALT 60 CALCULATION
      IF(E(I).GT.122..AND.E(I).LT.182.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.290..AND.E(I).LT.330.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.633..AND.E(I).LT.693.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.801..AND.E(I).LT.841.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.1144..AND.E(I).LT.1204.)CO60=CO60+IAREA2(I)
      IF(E(I).GT.1312..AND.E(I).LT.1352.)CO60=CO60+IAREA2(I)
C     ZINC 65 CALCULATION
      IF(E(I).GT.63..AND.E(I).LT.121.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.308..AND.E(I).LT.353.)ZN65=ZN65+IAREA2(I)
      IF(E(I).GT.574..AND.E(I).LT.632.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.841..AND.E(I).LT.864.)ZN65=ZN65+IAREA2(I)
      IF(E(I).GT.1085..AND.E(I).LT.1143.)ZN65=IAREA2(I)+ZN65
      IF(E(I).GT.1330..AND.E(I).LT.1375.)ZN65=ZN65+IAREA2(I)
C     POTASSIUM 40 CALCULATION
      IF(E(I).GT.438..AND.E(I).LT.478.)POT40=POT40+IAREA2(I)
      IF(E(I).GT.949..AND.E(I).LT.989.)POT40=POT40+IAREA2(I)
      IF(E(I).GT.1460..AND.E(I).LT.1500.)POT40=POT40+IAREA2(I)
  152 CONTINUE
C
C     THIS READS THE CALIBRATION VALUES AND CALCULATES
C     THE CONCENTRATION FOR EACH ISOTOPE PRESENT
C     THESE ARE MEASURED IN NANOCURIES
C
      VARIABLE='MNCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIMN=MN54*VALUE
      VARIABLE='CSCAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CICS=CS137*VALUE
      VARIABLE='OICAL'
      READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
      CIOI=OI131*VALUE
```

Figure B-13 (continued)

```
          VARIABLE='CRCAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CICR=CR51*VALUE
          VARIABLE='CO58CAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CIC58=CO58*VALUE
          VARIABLE='CO60CAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CIC60=CO60*VALUE
          VARIABLE='ZNCAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CIZN=ZN65*VALUE
          VARIABLE='POTCAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CIK40=POT40*VALUE
          VARIABLE='NACAL'
          READ(10,KEY=VARIABLE,KEYID=0)VARIABLE,VALUE
          CNA22=NA22*VALUE
C
C         THIS SECTION PRINTS THE RESULTS HEADING
C
          PRINT 100
          PRINT 159
          PRINT 160
          PRINT 161
          PRINT 162
          PRINT 176
          PRINT 163,NAME,DATE
          PRINT 164,ISSN
          PRINT 172,HEIGHT
          PRINT 173,WEIGHT
          IF(SEX.EQ.'F')PRINT 174,AGE
          IF(SEX.EQ.'f')PRINT 174,AGE
          IF(SEX.EQ.'M')PRINT 175,AGE
          IF(SEX.EQ.'m')PRINT 175,AGE
          PRINT 177
          PRINT 178
```

Figure B-13 (continued)

```
C
C       THIS PRINTS THE PEAK INFORMATION AFTER
C       DETERMINING IF THE PEAK IS WITHIN THE
C       THE 95% CONFIDENCE INTERVAL.
C
        PRINT 807
        DO 806 I=1,K
        SIGNIF=SIGMA*(SQRT(FLOAT(IAREA2(I))))
        IF(IAREA2(I).LT.SIGNIF)GO TO 806
        IF(IAREA2(I).GT.0)WRITE(6,808) E(I),IAREA2(I)
 806    CONTINUE
        PRINT 177
 807    FORMAT(15X,'THE FOLLOWING PEAKS WERE DETECTED:',//)
 808    FORMAT(1X,'ENERGY OF PEAK ',F10.2,' KEV.....
        1....AREA UNDER THE PEAK=',I8)
C
C       THIS PRINTS ONLY THOSE CONCENTRATIONS GREATER
C       THAN THE LOWER LIMIT FOR THE DETECTOR SYSTEM.
C
        PRINT 178
        IF(CICS.GT.0.)INDEX=INDEX+1
        IF(CICS.GT.0.)PRINT 165,CICS
        IF(CIOI.GT.0.)INDEX=INDEX+1
        IF(CIOI.GT.0.)PRINT 166,CIOI
        IF(CICR.GT.0.)INDEX=INDEX+1
        IF(CICR.GT.0.)PRINT 167,CICR
        IF(CIC58.GT.0.)INDEX=INDEX+1
        IF(CIC58.GT.0.)PRINT 168,CIC58
        IF(CIC60.GT.0.)INDEX=INDEX+1
        IF(CIC60.GT.0.)PRINT 169,CIC60
        IF(CIZN.GT.0.)INDEX=INDEX+1
        IF(CIZN.GT.0.)PRINT 170,CIZN
        IF(CIK40.GT.0.)INDEX=INDEX+1
        IF(CIK40.GT.0.)PRINT 180,CIK40
        IF(CNA22.GT.0.)INDEX=INDEX+1
        IF(CNA22.GT.0.)PRINT 182,CNA22
        IF(CIMN.GT.0.)INDEX=INDEX+1
        IF(CIMN.GT.0.)PRINT 183,CIMN
        IF(INDEX.EQ.0)PRINT 181
        PRINT 171
```

Figure B-13 (continued)

```
C
C       OUTPUT FORMAT STATEMENTS
C
 159    FORMAT(///,' ---------------------------------------
        1--------------------------')

 160    FORMAT(1X,'
        1 RENSSELAER POLYTECHNIC INSTITUTE ')

 161    FORMAT(1X,'
        1WHOLEBODY RADIATION COUNTING SYSTEM')

 162    FORMAT(1X,'------------------------------------
        1-----------------------------')

 163    FORMAT(//10X,A20,'WAS COUNTED ON',2X,A20)

 176    FORMAT(/15X,'THE FOLLOWING SUMMARY IS PROVIDED:')

 164    FORMAT(10X,'SOCIAL SECURITY ACCOUNT NUMBER'
        1,12X,I9)

 172    FORMAT(10X,'HEIGHT..................................',
        1I4,2X,'INCHES')

 173    FORMAT(10X,'WEIGHT..................................',
        1I4,2X,'POUNDS')

 174    FORMAT(10X,I2,'  YEARS OLD
        1    FEMALE',//)

 175    FORMAT(10X,I2,'  YEARS OLD
        1      MALE',//)

 177    FORMAT(1X,'------------------------------------
        1--------------------------')
```

Figure B-13 (continued)

```
178     FORMAT(15X,'QUANTITIES OF RADIOISOTOPES PRESENT:'
        1,//)

165     FORMAT(5X,'QUANTITY OF CESIUM 137 PRESENT   '
        1,F12.6,'  NANOCURIES')

166     FORMAT(5X,'QUANTITY OF IODINE 131 PRESENT   '
        1,F12.6,'  NANOCURIES')

167     FORMAT(5X,'QUANTITY OF CHROMIUM 51 PRESENT  '
        1,F12.6,'  NANOCURIES')

168     FORMAT(5X,'QUANTITY OF COBALT 58 PRESENT    '
        1,F12.6,'  NANOCURIES')

169     FORMAT(5X,'QUANTITY OF COBALT 60 PRESENT    '
        1,F12.6,'  NANOCURIES')

170     FORMAT(5X,'QUANTITY OF ZINC 65 PRESENT      '
        1,F12.6,'  NANOCURIES')

180     FORMAT(5X,'QUANTITY OF POTASSIUM 40 PRESENT'
        1,F12.6,'  NANOCURIES')

182     FORMAT(5X,'QUANTITY OF SODIUM 22 PRESENT    '
        1,F12.6,'  NANOCURIES')

183     FORMAT(5X,'QUANTITY OF MANGENESE 54 PRESENT'
        1,F12.6,'  NANOCURIES')

181     FORMAT(5X,'QUANTITY OF RADIATION PRESENT IS BELOW
        1 THE LOWER',/,' LIMIT OF DETECTION FOR THIS
        2SYSTEM.')

171     FORMAT(1X,'-------------------------------------
        1---------------------------',//)

        GO TO 99

999     CLOSE(10,STATUS='SAVE')
        CLOSE(12,STATUS='SAVE')
        RETURN 1
        END
```

APPENDIX C

COMPUTER INTERFACE MODULE

The following diagrams are the wiring diagrams for the interface module:

Figure C-1

First Stage Wiring Diagram

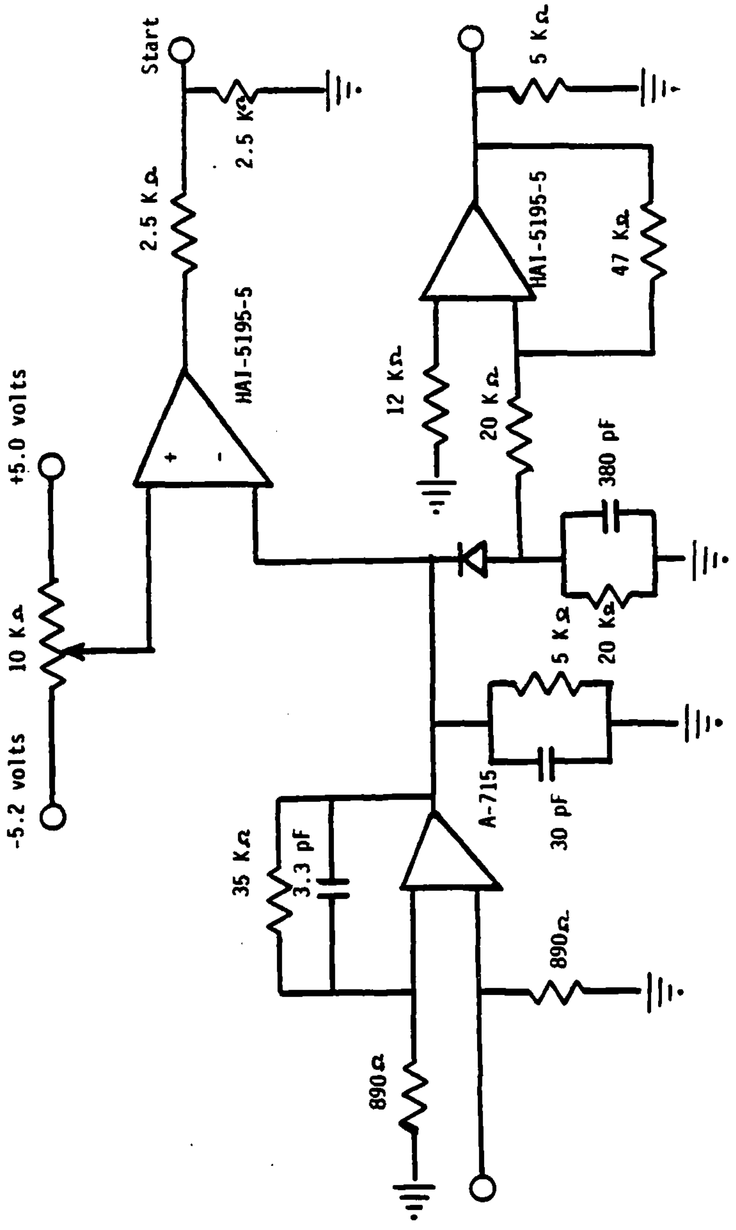


Figure C-1
Initial Amplification Circuit Diagram

Figure C-2

AM-6108 Analog to Digital Convertor
Pin Connections Schematic

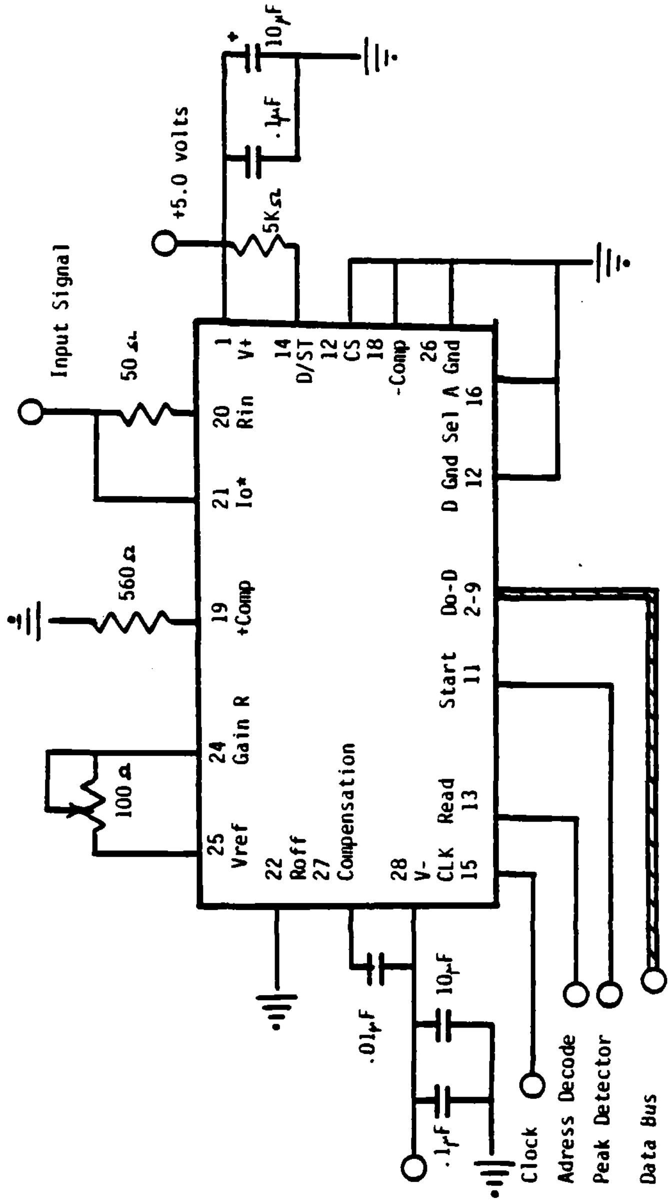


Figure C-2
Analog to Digital Converter Pin Connections

Figure C-3

10 MegaHertz Clock Circuit

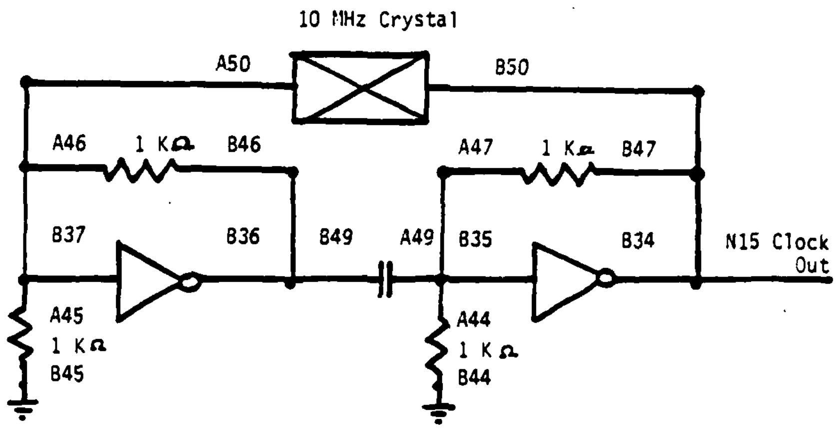
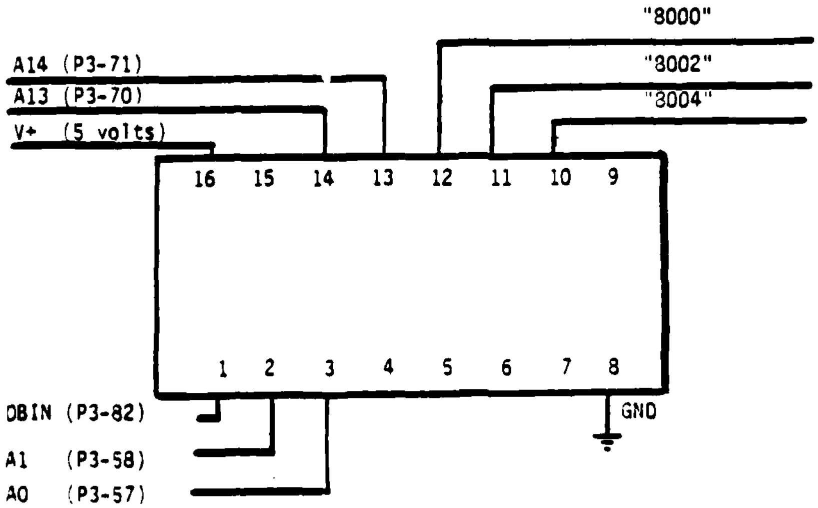


Figure C-4

Address Decoding (74LS139) Wiring Diagram



APPENDIX D

GAMMA INTERACTIONS

The following are examples of the type of in the type of interaction that occur within the Sodium Iodide Crystal

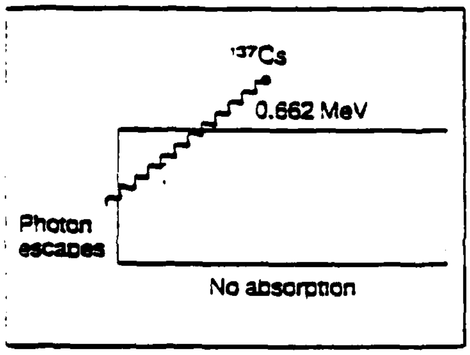
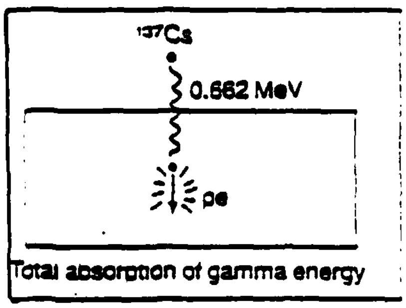


Figure D-1

No interaction • Energy absorbed by crystal = 0 MeV • No flash of light • No pulse



One photoelectric interaction • Energy absorbed by crystal = full energy of the incident gamma photon • One flash of light • Pulse counted under photopeak energy (= 0.662 MeV)

Figure D-2

*One Compton inter-
:tion with escape of
scattered photon •
:nergy absorbed by
:rystal = KE of Comp-
:on electron < 0.662
MeV • Flash of light
less bright than in
Figure 7b • Pulse
:ounted under ener-
gy lower than that
of the photopeak*

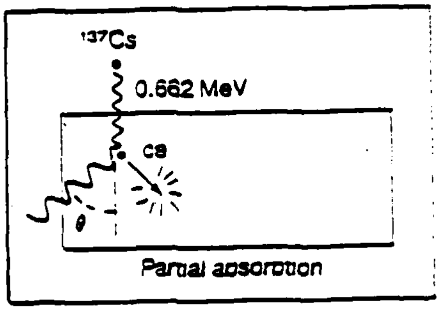
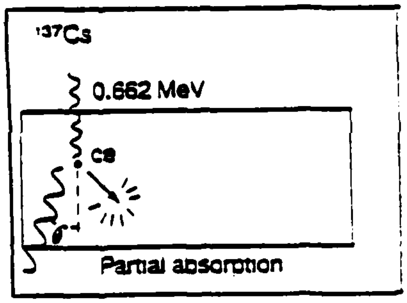


Figure D-3



*One Compton inter-
action with escape of
scattered photon •
Energy absorbed by
crystal < energy
absorbed in Figure 7c
(because of smaller θ)
• Flash of light less
bright than in Figure 7c
• Pulse counted under
energy lower than in
Figure 7c*

Figure D-4

*Two Compton inter-
:ctions followed by a
photoelectric inter-
action • Energy
:bsorbed by crystal =
0.662 MeV • Three
:shes of light almost
:multaneous • Inten-
:y of composite flash
: in Figure 7b • Pulse
counted under
photopeak*

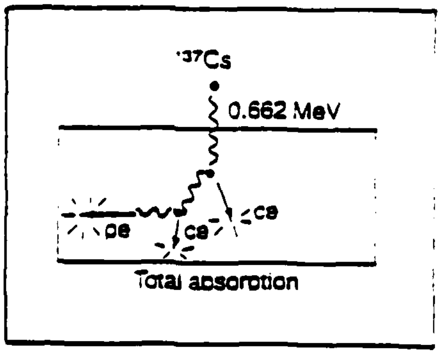
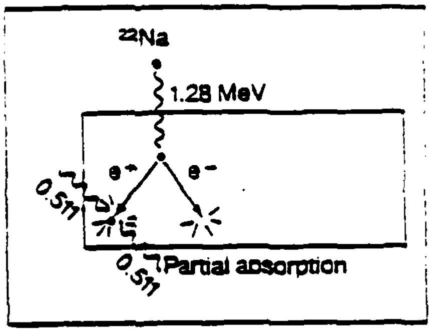


Figure D-5



Pair production with escape of both annihilation photons (double escape) • Energy absorbed by crystal = 1.28 MeV − 1.02 MeV = energy dissipated in crystal by electron and positron • Composite flash of light less bright than in Figure 7h • Pulse counted under energy = 1.28 MeV − 1.02 MeV

Figure D-7

Pair production with escape of one annihilation photon (single escape) • Energy absorbed by crystal = 1.28 MeV − 0.511 MeV = KE of pair + KE of photoelectron • Composite flash of light brighter than in Figure 7f but still less bright than in Figure 7h • Pulse counted under energy = 1.28 MeV − 0.511 MeV

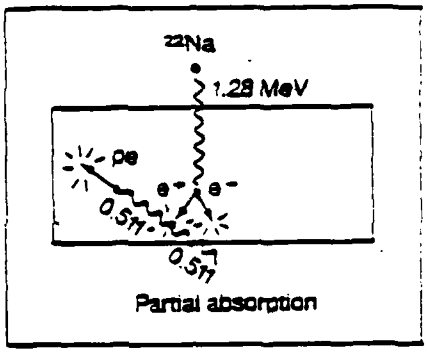
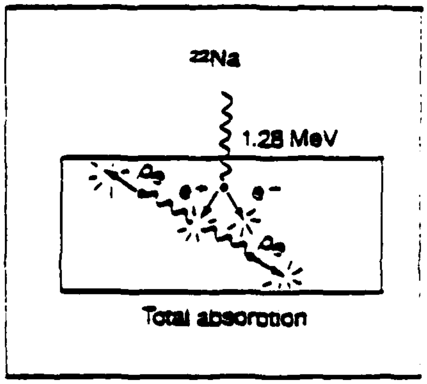


Figure D-8



Pair production with both annihilation photons absorbed • Energy absorbed by crystal = 1.28 MeV = KE of pair + KE of both photoelectrons • Composite flash is produced from intensity greater than in Figures 7f and 7g • Pulse counted under photopeak

Figure D-9

Legend:
pe, Photoelectron. ce, Compton electron. KE, Kinetic energy. e+, Positron. e−, Electron.

Reprinted from the Harshaw Equipment Catalog, 1983.

END

FILMED

8-85

DTIC