**VOLUME 1: TECHNICAL SPECIFICATIONS MANUAL
FOR THE
MARK-1 PULSED IONIZING RADIATION DETECTION SYSTEM**

**R. S. Lawrence
Y. D. Harker
J. L. Jones
J. M. Hoggan**

March 1993

**Westinghouse Idaho
Nuclear Company, Inc.**

**PREPARED FOR THE
DEPARTMENT OF ENERGY
IDAHO OPERATIONS OFFICE
UNDER CONTRACT DE-AC07-841D12435**

MASTER

TABLE OF CONTENTS

ABSTRACT	ix
1. INTRODUCTION	1
2. GENERAL SYSTEM DESCRIPTION	4
3. THEORY	10
3.1 Technical Basis	10
3.2 Unshielded Detector Response Versus Detector Thickness	13
3.3 Shielded Detector Response Versus Shield Thickness	19
3.4 Summary of the Theory	21
4. SYSTEM COMPONENT DESCRIPTION	22
4.1 MARK-1 Detection Unit	22
4.1.1 Power Supply System	22
4.1.2 Photon Detectors	29
4.1.3 Analog Front End Electronics	33
4.1.4 Logic Circuits	45
4.1.5 Operator Control	65
4.2 MARK-1 Support Unit	69
4.2.1 Portable Computer	69
4.2.2 Memory Card Reader	69
4.2.3 Printer	70
4.2.4 Cables	70
4.2.5 Documentation	72
4.2.6 Miscellaneous Supplies	72

5. SOFTWARE	73
5.1 Acquisition and Control Software	73
5.1.1 Software Development Kit	73
5.1.2 Data Acquisition and Control Software	85
5.2 Data Processing	90
5.2.1 Data Transfer	90
5.2.2 Data Analysis Routines	92
6. CALIBRATION OF THE MARK-1 SYSTEM	104
6.1 Introduction	104
6.2 Circuit Gains and Conversion Factors	106
6.3 Calibration Procedure	108
6.3.1 Equipment Required	108
6.3.2 Procedure	109

FIGURES

2.1	MARK-1 Radiation Detection System, Detection Unit and Support Unit	5
2.2	MARK-1 Detection Unit with Lid Open	6
2.3	MARK-1 System Block Diagram	7
2.4	MARK-1 Detector Orientation	8
3.1	Mass Attenuation Coefficient for Lead versus Photon Energy	11
3.2	Mass Attenuation Coefficient for BGO versus Photon Energy	12
3.3	Linear Attenuation Coefficient	18
4.1	Real-Time-Clock Block Diagram	24
4.2	Low Voltage Supply Block Diagram	25
4.3	High Voltage Supply Block Diagram	27
4.4	BGO/PMT Assembly	30
4.5	PMT Dynode Voltage Divider	31
4.6	Analog Front-End Block Diagram	34
4.7	MUX and ADC Block Diagram	35
4.8	Coincidence and Peak Detector Control Block Diagram	53
4.9	Timing Diagram - Peak Detector Control	54
4.10	Monitor/Control Panel	66
4.11	MARK-1 System Connections	71
6.1	Calibration Procedure Block Diagram	105
6.2	Calibration System Component Connections	115

TABLES

6.1	Calibration Data for MARK-1, System #1	122
6.2	Calibration Data for MARK-1, System #2	123

EQUATIONS

(3.1)		13
(3.2)		14
(3.3)		14
(3.4)		14
(3.5)		15
(3.6)		15
(3.7)		16
(3.8)		19
(5.1)		97
(5.2)		97
(5.3)		97
(5.4)		98
(5.5)		98

APPENDICES

APPENDIX A

Circuit Schematic Drawings	125
--------------------------------------	-----

Figures

A.1 Low Voltage Power Supply Schematic	126
A.2 Component Layout for Low Voltage Supply - Board #1	127
A.3 Component Layout for Low Voltage Supply - Board #2	128
A.4 High Voltage Power Supply Schematic	129
A.5 Component Layout for High Voltage Supply - Board #1	130
A.6 Component Layout for High Voltage Supply - Board #2	131
A.7 Analog Front-End Schematic	132
A.8 Component Layout for Analog Front-End Board	133
A.9 Logic Board - Page 1 of 4 Schematic	134
A.10 Logic Board - Page 2 of 4 Schematic	135
A.11 Logic Board - Page 3 of 4 Schematic	136
A.12 Logic Board - Page 4 of 4 Schematic	137
A.13 Component Layout for Logic Board	138
A.14 Monitor/Control Panel Schematic	139

APPENDIX B

Software Flowcharts and Listings	140
--	-----

APPENDIX B.1

Acquisition and Control Software Flowcharts	141
---	-----

Figures

B.1 Flowchart for: Acquisition Program MAIN()	142
B.2 Flowchart for: Acquisition Function EVENT1	143
B.3 Flowcharts for: DUMP(), STORE(), RDET	144
B.4 Flowcharts for: RDTEMP(), RDCLK(), STGN()	145
B.5 Flowchart for: SETGAIN	146

APPENDICES (continued)

B.6	Flowchart for: CAL()	147
B.7	Flowchart for: SELGN()	148
B.8	Flowchart for: SETCLK()	149
B.9	Flowchart for: READ()	150
B.10	Flowcharts for: RXCHAR(),CHGHV(), TXCHAR(), TXCHK()	151

APPENDIX B.2

Analysis Software Flowcharts	152
--	-----

Figures

B.11	Flowcharts for: Analysis MAIN, SEL SYS/FILE, DISP RATIO GRAPH .	153
B.12	Flowcharts for: PRINT RATIO DATA, CALL RATIO PLOTS, PROCESS 'A' GRAPH	154
B.13	Flowchart for: DRAW SLOPE LINE	155
B.14	Flowcharts for: PROCESS 'B' GRAPH, PRINT RATIO DATA-NO GRAPH	156
B.15	Flowchart for: COMPUTE COMBINED DATA	157
B.16	Flowcharts for: READ, CONVERT ASCII TO DECIMAL	158
B.17A	Flowchart for: DRAW GRAPH - Part 1 of 2	159
B.17B	Flowchart for: DRAW GRAPH - Part 2 of 2	160
B.18	Flowcharts for: FIT, PRINT SLOPE AND INTERCEPT-NO GRAPH . . .	161
B.19	Flowchart for: COMPUTE PHA	162
B.20A	Flowchart for: PRINT PHA Part 1 of 2	163
B.20B	Flowchart for: PRINT PHA Part 2 of 2	164

APPENDIX B.3

Acquisition and Control Program Listings	165
--	-----

APPENDIX B.4

Analysis Program Listing	189
------------------------------------	-----

ABSTRACT

The MARK-1 detection system was developed by the Idaho National Engineering Laboratory for the U.S. Department of Energy Office of Arms Control and Nonproliferation. The completely portable system was designed for the detection and analysis of intense photon emissions from pulsed ionizing radiation sources. This manual presents the technical design specifications for the MARK-1 detection system and was written primarily to assist the support or service technician in the service, calibration, and repair of the system. The manual presents the general detection system theory, the MARK-1 component design specifications, the acquisition and control software, the data processing sequence, and the system calibration procedure. A second manual entitled: *Volume 2: Operations Manual for the MARK-1 Pulsed Ionizing Radiation Detection System* (USDOE Report WINCO-1108, September 1992) provides a general operational description of the MARK-1 detection system. The Operations Manual was written primarily to assist the field operator in system operations and analysis of the data.

1. INTRODUCTION

The MARK-1 radiation detection system was developed at the Idaho National Engineering Laboratory (INEL) and supported by the Department of Energy (DOE) Office of Arms Control and Nonproliferation (OACNP) as a method of detecting and identifying key characteristics of high-intensity, pulsed high-energy X-ray or gamma-ray sources. The specific design requirements for the detection system include:

- detection capability of high-energy or high-intensity photons from pulsed bremsstrahlung or fissioning sources,
- identification or characterization of the photon energy emissions for each detected source pulse,
- measurement and storage of source pulse data,
- differentiation between pulsed source photons and random background noise,
- self-activation (i.e., triggering) mechanism,
- broad sensitivity range,
- remote operation capability (i.e., self-contained power supply), and
- one person portability.

The MARK-1 design evolved from two earlier designs: a proof-of-principle system and a prototype unit. The proof-of-principle system was constructed from off-the-shelf electronic modules, was transportable via a light utility vehicle, and required a portable electric generator to meet the power requirements. Following the proof-of-principle design, a prototype unit was developed which considerably down-sized the proof-of-principle design and still optimized and/or met many of the above stated requirements. After successful

testings of the prototype unit, the MARK-1 radiation detection system was developed which met all the above design criteria.

The MARK-1 is a field-deployable, portable, battery-operated, remote-operated system intended for the detection of ionizing radiation from pulsed ionizing radiation sources for finite periods of time. The system was designed to detect, measure, and analyze the photon energy distribution for each detected source pulse response. A typical source pulse is produced when a high-energy charged particle beam strikes a high atomic number converter target or interacts directly in air. During this media interaction, each source pulse will result in the generation of a high intensity photon production event called a "gamma flash." The characteristics of the "gamma flash" are determined by a number of factors including such parameters as beam pulse width, photon intensity (which is related to the beam current), and particle beam energy.

There are two manuals prepared for servicing, calibrating and operating the MARK-1 detection system. This manual, Volume 1, is intended to be used primarily for service calibration, and repair; whereas the second manual, Volume 2, is intended to be used primarily for field operations. Two levels of technicians have been identified to support the MARK-1 system; i.e., the support technician and the field operator. No special prerequisite training is required of the field operator other than a thorough understanding of Volume 2 and a basic familiarity with minicomputer operation and the Microsoft Disk Operating System (DOS). The support technician must not only be a qualified field operator, but must also have sufficient electronics knowledge to understand the system's fundamental operations theory, to interpret the system's electronic (circuit) diagrams and descriptions, to comprehend the details of the acquisition software and the software's physical interaction with the hardware, and to perform any and all necessary system repairs and upgrades. Volume 1 of

this series contains the technical specifications for the MARK-1 detection system along with instructions for the support technician. Volume 2 contains the technical information and instructions for the field operator.

Each volume provides a description of the two hardware units comprising the complete MARK-1 system and identifies all major functional components in each unit. The general description section is followed with descriptive details of the component-to-component operational overview of the system and major components that require operator interaction. In Volume 1, the system description section is followed by a detailed description of the detector systems, control electronics and software routines used in acquisition control and data analysis. This section is written in sufficient detail to enable a designer or a support technician to diagnose, calibrate, and repair the system. In Volume 2, the general system description section is followed by sections providing detailed step-by-step procedures for hardware setup, initialization, operation, and post-acquisition data processing analysis and interpretation. A section describing the procedure for calibration checking is included; however the actual calibration procedure is included in Volume 1 and should only be attempted by a qualified support technician.

2. GENERAL SYSTEM DESCRIPTION

The MARK-1 system, shown in Figure 2.1 consists of two portable, suitcase-sized units: the detection unit and the support unit. A view of the opened detection unit is shown in Figure 2.2. The detection unit is a battery-operated system that requires no external support for a period up to 18 hr, can acquire pulsed data at a pulse rate up to 60 Hz, and contains the radiation detectors, dc power supplies, analog circuits, system microcontroller, and data memory card. The detection unit suitcase is 53 cm (21") long, 33 cm (13") tall, and 16.5 cm (6.5") wide and weighs approximately 45 lb. The main functional components of the detection unit are shown in a simplified block diagram in Figure 2.3. The support unit suitcase, is used only for initial system setup and data analysis. The support unit is 53 cm (21") long, 43 cm (17") tall, and 18 cm (7") wide and weighs approximately 35 lb. The suitcase contains a portable off-the-shelf laptop computer, a battery-powered printer, a commercially available memory card reader, associated system cables, and related operating manuals.

Within the MARK-1 detection unit are eight independent cylindrical bismuth germinate (BGO) scintillation crystal detectors oriented as shown in Figure 2.4. Each BGO crystal is coupled to a photomultiplier tube (PMT) with a charge-sensitive preamplifier, shaping amplifier, discriminator, and peak detection circuit. Four of the eight detectors are vertically oriented in the upright suitcase configuration. The vertically oriented detectors have unshielded crystals having a fixed length of 2.54 cm and various diameters (0.64, 1.23, 1.84, and 2.40 cm). The crystals in the other four detections are horizontally oriented in the upright suitcase configuration, with a fixed length of 2.54 cm and a fixed diameter of 2.40 cm, and are surrounded by lead shielding having thicknesses of 0.1016, 0.5080, 1.0160, and 1.524 cm.

MARK-1: RADIATION DETECTION SYSTEM OPERATIONS MANUAL

Figure 2.1 The MARK-1 pulsed ionizing radiation detection system consists of two suitcases; the smaller case is the field-deployable detection unit and the larger case contains the support equipment needed to off-line read and process

Figure 2.2 MARK-1 Detection Unit with Lid Open.

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

General System Description

Figure 2.3 MARK-1 System Block Diagram.

MARK-1 DETECTOR CONFIGURATION

Figure 2.4 MARK-1 Detector Orientation.

The MARK-1 detection unit detects and measures the intensity of a pulsed source event, or "gamma flash," by measuring the total energy deposited in a scintillation crystal by photon interactions (i.e., photoelectric effect, Compton scattering, and pair production). System activation or triggering is accomplished by simultaneous coincident event detection of the high intensity "gamma flash" in multiple detectors. Since the background noise intensity is orders of magnitude less than the "gamma flash" intensity, the probability of simultaneous events occurring from a random background is quite low. For the MARK-1 system, the number and specific combination of detectors required for a coincidence is selected by the operator.

Upon detection of a coincidence event, the system microcontroller reads the peak intensities from the eight detectors and stores them in memory along with the date and time of the event (to the nearest millisecond), two temperature readings, and amplifier gain information to allow post-processing analyses. All data are stored in a removeable, "credit card" size memory cartridge. After a data acquisition period, the memory cartridge is removed from the detection unit and read using a portable laptop computer. The differences in detector response resulting from the differences in detector size and shield thickness yield information characteristic of the pulsed source. Analyses of the stored pulsed data provide the total energy deposited per pulse, dose per pulse, dose rate over the collection time, source pulse rate, and source stability. The time for reading a memory cartridge and analyzing its data is approximately 10 minutes.

3. THEORY

The MARK-1 detection system was designed to characterize high-intensity, pulsed source photon emissions via the determination of average energies of the photon pulse. Since many photons arrive at a detector during a pulsed event, conventional gamma-ray spectrometry involving pulse-height analysis of single photon events cannot be used to these measurements. This section describes the basic detection system theory.

3.1 Technical Basis

The physical phenomenon on which the measurements of the MARK-1 system is based is the energy dependence of the photon attenuation in various thicknesses of lead and BGO. Figures 3.1 and 3.2 show the linear attenuation coefficients versus photon energy for lead and BGO, respectively. The lower-energy portions of these curves are dominated by the photoelectric effect; the intermediate-energy portion is a combination of photoelectric and Compton scattering effects; and the higher-energy portion is a combination of Compton scattering and electron-positron pair production. In the MARK-1 system, the detector response is a pulse whose amplitude is proportional to the total energy produced by multiple photon interactions in the detector. The photon energy spectrum is obtained by measuring the detector pulse amplitude as a function of detector thickness and as a function of surrounding shield thickness. From the detector response versus detector size, information is obtained about the lower energy portions of the photon spectrum. From the response versus shield thickness, information is obtained about the higher energy portions of the photon spectrum.

Linear Attenuation Coefficients for Pb

R93 0238

Figure 3.1 Mass Attenuation Coefficient for Lead versus Photon Energy.

Linear Attenuation Coefficients for BGO

R93 0239

Figure 3.2 Mass Attenuation Coefficient for BGO versus Photon Energy.

3.2 Unshielded Detector Response Versus Detector Thickness

The MARK-1 detection unit employs four unshielded BGO detectors each having a fixed length of 2.54 cm and various diameters ranging from 0.64 to 2.40 cm. Due to the vertical orientation (see Figure 2.4) of these detectors in an upright suitcase (i.e., a deployed configuration) the diameter variations correspond to the detector thickness variations. For the purposes of this discussion, assume that the source-to-detector distance is large and that the photon field can be treated as a parallel beam incident on the detectors. Also assume the a detector can be represented by a rectangular volume with a characteristic cross sectional area, A , normal to the photon beam and a thickness, t_d parallel to the photon beam. The dose in the detector material as a function of photon energy, E , and position in the detector, x , is given as:

$$D(E,x) = C [\mu_e(E)/\rho] E \phi(E,x) \quad (3.1)$$

where: $D(E,x)$ = Dose per unit photon energy in the detector material (cGy per MeV),

E = photon energy (MeV),

x = the distance into detector (m),

C = conversion constant

(for the units shown $C = 1.602 \times 10^{-11}$ Cgy·Kg·MeV⁻¹),

$\mu_e(E)$ = energy attenuation coefficient (m⁻¹),

ρ = the density of the detector material (Kg·m⁻³),

$\phi(E,x)$ = photon fluence per unit energy (m⁻²·MeV⁻¹),

The photon fluence per unit photon energy is given by:

$$\phi(E,x) = \phi_o(E) \exp(-\mu_a x) \quad (3.2)$$

where: $\phi_o(E)$ = the photon fluence per unit photon energy incident on the detector,
 μ_a = the total absorption attenuation coefficient for the detector material.

The total energy deposited in the crystal, as a function of the photon energy (E) and detector thickness (t_d), is an integral of the dose over the volume of the detector.

$$W(E,t_d) = A \int_0^{t_d} \rho D(E,x) dx \quad (3.3)$$

where: $W(E,t_d)$ = the total energy per unit photon energy deposited in the detector,
 A = cross sectional area of the detector,
 and t_d = the detector thickness.

Solving the integral and substituting for $D(E,x)$ yields,

$$W(E,t_d) = C A \rho \frac{\mu_e(E)}{\rho} E \phi_o(E) \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)} \quad (3.4)$$

The average dose in the detector material of a given thickness is equal to the total energy deposited in the detector divided by the mass of the detector.

$$D(E, t_d) = \frac{\mu_e(E)}{\rho} \phi_o(E) \frac{1 - \exp(-\mu_a t_d)}{\mu_a(E) t_d} \quad (3.5)$$

where: $D(E, t_d)$ = average dose per unit photon energy in the detector volume as a function of photon energy, E , and detector thickness, t_d .

To arrive at the total dose for all photon energies, equation 3.5 is integrated over the photon energy from 0 to maximum photon energy, $E_{\max}$. Because the measured detector response is analyzed in terms of a power series in t_d , equation 3.5 is likewise expanded in the power series.

$$D(E, t_d) = C \mu_e(E) E \phi_o(E) \left[1 - \frac{1}{2} \mu_a(E) t_d + \frac{1}{6} \mu_a(E)^2 t_d^2 - \frac{1}{24} \mu_a(E)^3 t_d^3 + * * * \right] \quad (3.6)$$

The result of integrating equation 3.6 over energy is another power series in t_d , where the coefficients contain the integrals over energy as shown in equation 3.7.

$$D(t_d) = C \left[\langle 0 \rangle - \frac{1}{2} \langle 1 \rangle t_d + \frac{1}{6} \langle 2 \rangle t_d^2 - \frac{1}{24} \langle 3 \rangle t_d^3 + * * * \right] \quad (3.7)$$

where: $D(t_d)$ = average dose in a detector of thickness, t_d , integrated over all photon energies,

$E_{\max}$ = upper energy point of the photon spectrum,

$$\langle 0 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} E \phi_o(E) dE \quad ,$$

$\langle 0 \rangle$ is the weighted 0th order moment of $\mu_a(E)$,

$$\langle 1 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} E \phi_o(E) \mu_a(E) dE \quad ,$$

$\langle 1 \rangle$ is the weighted 1st order moment of $\mu_a(E)$,

$$\langle 2 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} \phi_o(E) \mu_a(E)^2 dE \quad ,$$

$\langle 2 \rangle$ is the weighted 2nd order moment of $\mu_a(E)$,

$$\langle 3 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} \phi_o(E) \mu_a(E)^3 dE \quad ,$$

$\langle 3 \rangle$ is the weighted 3rd order moment of $\mu_a(E)$,

$$\begin{matrix} \cdot & \cdot \\ \cdot & \cdot \end{matrix}$$

In Equation 3.7, the coefficient of each t_d^n term contains a weighted n^{th} order moment of the absorption coefficient; i.e., $\langle 0 \rangle$, $\langle 1 \rangle$, $\langle 2 \rangle$, $\langle 3 \rangle$, etc., where the weighting function is:

$$w(E) = \frac{\mu_e(E)}{\rho} E \phi_o(E) \quad .$$

The ratio of 1st moment to the 0th moment, $\langle 1 \rangle / \langle 0 \rangle$, is the weighted average absorption coefficient, the ratio of the 2nd moment to the 0th moment, $\langle 2 \rangle / \langle 0 \rangle$, is likewise the weighted average of the square of the absorption coefficient. The difference between the squared average and the average squared, $(\langle 2 \rangle / \langle 0 \rangle) - (\langle 1 \rangle / \langle 0 \rangle)^2$, is the variance of the absorption coefficient over the energy distribution. The square root of the variance is termed the standard deviation. In this application, the range from the average attenuation coefficient minus the standard deviation to the average attenuation plus the standard deviation is an effective range of absorption coefficient values. This average and range in attenuation coefficients are converted to an effective photon energy and range by using the attenuation coefficient versus photon energy as shown in Figure 3.3. In this figure, the weighted average and range are shown as dashed horizontal lines, μ_{eff} , $\mu_L - \mu_U$, respectively. The corresponding energy values are shown as dashed vertical lines, E_{eff} , $E_L - E_U$, respectively.

Linear Attenuation Coefficient

R93 0253

Figure 3.3 Linear Attenuation Coefficient.

The higher order moments are able to provide more detailed information about the photon spectrum and in principle the photon spectrum can be generated from an infinite set of moment results. However, because of the detector sizes and the lead shield thicknesses used in the MARK-1 system, the practical limit is second order.

3.3 Shielded Detector Response Versus Shield Thickness

In addition to the unshielded detectors, the MARK-1 system has four shielded detectors where the thickness of the lead shield is varied from 0.1 cm to 1.5 cm.

Using the same derivation approach applied to the unshielded detector response versus detector thickness, a power series expansion of the dose versus shield thickness is obtained as shown in Equation 3.8.

$$D(t_d, t_s) = C \left[\langle 0 \rangle_s - \langle 1 \rangle_s t_s + \frac{1}{2} \langle 2 \rangle_s t_s^2 - \frac{1}{6} \langle 3 \rangle_s t_s^3 * * * \right] \quad (3.8)$$

where: $D(t_d, t_s)$ = average dose in the detector as function of detector thickness, t_d , and shield thickness, t_s ,

C = the units conversion constant (same as in unshielded detector response),

t_s = shield thickness,

$\mu_s(E)$ = attenuation coefficient of the shield material,

$$\langle 0 \rangle_s = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} E \phi_o(E) \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)t_d} dE$$

$\langle 0 \rangle_s$ is the weighted 0th order moment of $\mu_s(E)$

$$\langle 1 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} E \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)t_d} \mu_s(E) dE ,$$

$\langle 1 \rangle_s$ is the weighted 1st order moment of $\mu_s(E)$,

$$\langle 2 \rangle = \int_0^{E_{\max}} \frac{\mu_e(E)}{\rho} E \phi_o(E) \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)t_d} \mu_s(E)^2 dE ,$$

$\langle 2 \rangle_s$ is the weighted 2nd order moment of $\mu_s(E)$,

$$\langle 3 \rangle = \int_0^{E_{\max}} \mu_e(E) E \phi_o(E) \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)t_d} \mu_s(E)^3 dE ,$$

$\langle 3 \rangle_s$ is the weighted 3rd order moment of $\mu_s(E)$,

·
·
·

The weighting function for the moments of the shield attenuation coefficient is:

$$w_s(E) = \frac{\mu_e(E)}{\rho} E \phi_o(E) \frac{1 - \exp(-\mu_a(E)t_d)}{\mu_a(E)t_d}$$

As in the case of the unshielded detector response versus detector thickness, the shielded detector response versus shield thickness provides another set of attenuation coefficient moments and effective photon energy and range. Because of the different weighting function and different material, the shielded parameters will be different from the unshielded parameters.

3.4 Summary of the Theory

As explained in detail in the Section 3.2, the changes in detector response with increasing detector thickness were used to calculate an average and range of attenuation coefficients and from these an average and range of effective photon energies were inferred. In a similar manner, as explained in Section 3.3, the change in detector response with increasing shield thickness is used to calculate another set of average and range of attenuation coefficients and an average and range of effective photon energies. By comparing both sets of values with corresponding calibration values obtained by using standard photon sources and calculated values for other sources, the characteristics of the photon source can be determined.

4. SYSTEM COMPONENT DESCRIPTION

The MARK-1 system consists of two separate units: the detection unit and the support unit. The detection unit provides all the photon detection and data storage capability of the system. The second unit contains all the hardware needed to support the detection unit during initial deployment and post-processing analyses. This section describes in detail the various hardware components of the MARK-1 system. Block diagrams are included to aid in the descriptions. Detailed circuit schematics are included in Appendix A of this manual.

4.1 MARK-1 Detection Unit

4.1.1 Power Supply System

The MARK-1 power system consists of a low voltage supply and a high voltage supply supported by several batteries.

4.1.1.1 Batteries The MARK-1 system utilizes a total of six batteries to provide electric power: one 12-V battery in the detection unit provides the main system power, three 1.5-V batteries on the detection unit's logic board supply holding power, and two 9-V batteries in the remote control box enable remote operation.

The main power source for system operation is a 12-V, 12-A-hour, lead-acid rechargeable battery. The MARK-1 system draws about 670 Ma resulting in about 18 hr of continuous operation from a fully charged battery. The main battery is installed on the left side of the detection unit suitcase directly below the logic board (see Figure 2.2). It is

charged by an external, ac powered charger unit. The charger is connected to the battery through a Lemo connector on the outside of the detector unit suitcase.

Three size "AA," 1.5-V, alkaline batteries are used on the logic board. During periods when the main system power is in an OFF state, these batteries provide the holding or backup power to the real-time-clock (RTC), a programmable gate array (PGA2) in the microcontroller circuit, and the automatic gain latch chip in the high voltage supply (see Section 4.1.1.3). The real-time-clock and battery backup are shown schematically in Figure 4.1 and described in Section 4.1.4.2. Holding power to the PGA2 is needed to assure proper microcontroller operation.

Two 9-V, series connected, alkaline batteries are used in the remote control box to turn the system power ON/OFF from a remote location. The remote box controls a latching relay installed in the MARK-1 Monitor/Control Panel. This relay is mechanically latched in either the open or closed position with the application of a momentary voltage signal. No battery power is required to hold the relay. The relay contacts are connected in parallel with the monitor/control panel main power switch. In the OPEN position, system power is applied through the on-board main power switch. In the CLOSED position, system power is applied through the relay contacts. (See the Low Voltage Supply Block Diagram, Figure 4.2, and Low Voltage Power Supply Schematic, Appendix Figure A.1.)

4.1.1.2 Low Voltage System The low voltage supply is located in the upper right corner of the detection unit suitcase (see Figure 2.2). A block diagram of this power supply is presented in Figure 4.2. This module is powered by the 12-V main battery. A blocking diode is used to allow in-circuit charging of the battery. Commercially available dc/dc converter modules are used to generate the output voltages. The output voltages of these

**BLOCK DIAGRAM - REAL TIME CLOCK
AND BACKUP BATTERY**

Figure 4.1 Real-Time-Clock Block Diagram.

BLOCK DIAGRAM - LOW VOLTAGE SUPPLY

Figure 4.2 Low Voltage Supply Block Diagram.

modules are ± 12 Vdc and ± 5 Vdc. Each output voltage is fed through a series field effect transistor (FET) controlled by the 5-Vdc outputs. The positive 5-Vdc output controls the positive output FET's. The negative 5-Vdc output controls the negative output FETs. The purpose for the FET control is to prevent the ± 12 -Vdc outputs from being applied to the system before the ± 5 -Vdc outputs to prevent damage to the DG538 analog multiplexer chips.

The output voltages from the FETs are filtered by resistor/inductor/capacitor (RLC) filter networks to minimize noise on the system power lines. Potential sources of noise are the DC/DC converters and switching spikes in the high voltage supply. (See also the Low Voltage Power Supply Schematic, Appendix Figure A.1, and the circuit board component layout in Figures A.2 and A.3.)

4.1.1.3 High Voltage Supply The high voltage supply module is located near the top center of the detection unit suitcase. This supply is the unit immediately to the right of the logic board and above the detectors as seen in Figure 2.2. It provides a negative high voltage to the eight photomultiplier tubes (PMT's). The High Voltage Supply Block Diagram is shown in Figure 4.3, the detailed schematic in Appendix Figure A.4, and the printed circuit board layout in Figures A.5 and A.6.

The main component in this supply is the DC/DC high voltage module produced by Venus Scientific Corporation. Power for the Venus Scientific high voltage module is obtained through a 2N2219 NPN transistor from the 12-V main battery. This module is capable of a 1500-Vdc output and 1 mA. It is part of a non-inverting, operational amplifier (OP AMP) type configuration, with the module installed inside the OP AMP loop. As shown in the upper right portion of Figure A.4 of Appendix A, the high voltage output of the OP AMP configuration is fed through a filter network (see elements R29, R30, and C3-C9)

BLOCK DIAGRAM - HIGH VOLTAGE SUPPLY
WITH AUTOMATIC GAIN

Figure 4.3 High Voltage Supply Block Diagram.

before being applied to the eight PMTs or sent to an analog multiplexer. The filter network minimizes switching spike noise from the high voltage power supply module (Venus Scientific module K15).

The feedback network for the OP AMP configuration is a 1:100 voltage divider, as shown in Figure 4.3, which supplies 10 mV of feedback per 1 V of output. The feedback network is a resistor divider string containing twenty 1-M Ω resistors in the feedback network with a 182-k Ω resistor in series with a user-selectable resistor to ground at the negative input. The user selectable resistor results in potential gains of approximately 87 to 111. The procedure for determining the value of the selectable resistor is given in the MARK-1 Calibration Procedure (see Section 6).

Before high voltage is applied to any PMT, the filtered high voltage is supplied to eight 2.5-M Ω potentiometers operating in parallel. These potentiometers are used to adjust the individual high voltages supplied to the eight PMT's. The output voltages from the potentiometers are fed to the individual detector assemblies via eight Lemo connectors on the outside of the high voltage module.

The filter high voltage is also fed to a 1:2000 resistor divider string followed by an LF411 OP AMP with a gain of two. The resulting output from the LF411 OP AMP is a voltage level which is 1:1000 of the high voltage output from the filter network. The reduced voltage is supplied to the control panel high voltage monitor point and to an analog multiplexer on the logic board. The microcontroller reads the reduced high voltage signal from the multiplexer and stores that value in the current data record.

The output voltage of the module is controlled by a negative control voltage output from a digital-to-analog converter (DAC). The input to the DAC is a digital control word generated by the on-board microcontroller (see Section 4.1.4.4). The information to generate the control word is obtained from an AUTO GAIN LATCH in the microcontroller during start-up. As previously mentioned, batteries are required to provide holding power to this automatic gain latch (Number 74HC573) since it provides certain start-up gain information to automatically preset the high voltage supply to a previously selected output.

4.1.2 Photon Detectors

The photon detectors in the MARK-1 detection unit use bismuth germanate (BGO) scintillation crystals coupled to photomultiplier tubes (PMT). Each BGO crystal is cylindrical in shape with a length of 2.54 cm. The PMT for each detector assembly is attached to the end of the BGO crystal by an optical coupling grease. Each BGO/PMT assembly, along with the dynode voltage divider network is packaged in a cylindrical aluminum housing and sealed. A typical assembly is shown in Figure 4.4. The assembly case is grounded to minimize electrical pickup and interference. In addition, a mu-metal shield is wrapped around the outside of the assembly to eliminate interference from any potential magnetic fields. There are eight detector assemblies in each MARK-1 detection unit.

A schematic diagram of the PMT and dynode voltage divider network is shown in Figure 4.5. The PMT's are Hamamatsu Model R1924. They are cylindrical in shape, 2.54 cm diameter x 5.6 cm long (this length includes the length of electrical connection pins). There are 10 dynodes in each PMT. The nominal current amplification is 4.4×10^5 at 1000-V anode-to-cathode voltage. Nominal operating voltages in the MARK-1 detection

CONFIGURATION DRAWING FOR PHOTOMULTIPLIER/DETECTOR ASSEMBLIES
WITH INTEGRATED VOLTAGE DIVIDER STRING AND PMT WITH FLEXIBLE LEADS

Figure 4.4 BGO/PMT Assembly.

Figure 4.5 PMT Dynode Voltage Divider.

unit vary, from 350 to 750 V depending on the sensitivity range established by the microcontroller. This produces current amplifications from 2×10^2 to 4.5×10^4 . The polarity of the PMT high voltage is: anode at ground potential and the cathode at the negative high voltage. The anode-to-cathode voltage is applied across the PMT voltage divider network to achieve the specified voltage level at each dynode in the PMT. The first element in the PMT voltage divider network consists of a 2.5-M Ω variable resistor in series with a 1.5-M Ω resistor connected between the cathode and first dynode. The next element is a 1-M Ω resistor connected between first and second dynodes. The 1-M Ω resistor is repeated between adjacent dynodes through dynode #10 and from dynode #10 to ground. Dynodes #8, #9, and #10 also have decoupling capacitors to minimize drifts in dynode voltages under conditions of high signal currents (i.e., signal current comparable to divider network current).

An RG-174 coaxial cable is used to connect each PMT cathode to the corresponding output from the high voltage module. The detector response signal from each PMT anode is carried over another RG-174 coaxial cable to the corresponding analog circuit input. Each cathode-to-first-dynode 2.5-M Ω variable resistor is located on the outside of its BGO/PMT assembly. This allows adjustment of this resistor without disassembling the PMT assembly. Two #22-gauge wires connect the external variable resistor to the remainder of the PMT voltage divider network located inside the BGO/PMT case.

The MARK-1 detection unit output relies on quantitative comparisons of the peak signal responses from the eight detector assemblies in the detection unit. In order to accomplish this type of comparison, the gains of the detector PMT's must be matched at all sensitivity ranges used in the MARK-1 detection unit and a graph of gain versus high voltage is measured for each PMT. Because of small differences in the construction of the individual

PMTs, their gains will be mismatched at any given high voltage setting. To compensate for gain mismatch, each individual PMT has its own potentiometer in the high voltage supply circuit. Therefore, the high voltage applied to each PMT can be individually adjusted to achieve gain matching. For details about the high voltage system the reader is referred to Section 4.1.1.3. Individual adjustment of the PMT high voltage is able to achieve gain matching at only one particular gain setting; however, gain matching must be accomplished at all gain settings used in the MARK-1 system. This requires that the slope of the gain versus high voltage curve for each PMT matches the slopes of the gain versus voltage curves for the other PMT's in the detection unit. For small differences in slopes, the 2.5-M Ω variable resistor between the cathode and first dynode can be adjusted to achieve slope matching. However, this adjustment also affects PMT gain, so an adjustment for slope also entails a readjustment of the high voltage to achieve the same gain. In order to achieve optimal gain and slope matching between PMT's, a series of iterative adjustments of each PMT high voltage (i.e., gain) and cathode-to-first-dynode resistance (i.e., slope and gain) will be required. The details of this adjustment process are covered in the Calibration Procedure in Section 6.

4.1.3 Analog Front-End Electronics

The description of the MARK-1 detection unit analog front-end electronics includes all functions that involve analog signal processing. Specifically, these functions are: "Analog Front-Ends" (AFEs), "Analog Multiplexers" (MUXs), and "Analog-to-Digital Converters" (ADCs). Simplified block diagrams of these networks are shown in Figure 4.6 for the AFE and Figure 4.7 for the MUX and ADC. In addition, the schematic drawing for the AFE circuit is shown in Appendix Figure A.7, and Appendix Figure A.9 for the MUX and ADC

BLOCK DIAGRAM - ANALOG FRONT END
(TYPICAL FOR 8 CHANNELS)

Figure 4.6 Analog Front-End Block Diagram.

BLOCK DIAGRAM - ANALOG MULTIPLEXERS (MUX) AND
ANALOG-TO-DIGITAL CONVERTER (ADC)

Figure 4.7 MUX and ADC Block Diagram.

functions. Figures A.8 and A.13 show the printed circuit board component locations for these circuits.

The block diagram for the AFE shows the main features of this circuit for one detector. There are eight such circuits and each circuit is installed in an individual amplifier box. The input signal to an AFE is the output signal from the corresponding BGO/PMT detector assembly. The first stage in the AFE is a charge-sensitive preamplifier. The output signal from the preamplifier is fed to the high-gain fast amplifier, the shaping amplifier circuit, and an optional high-gain amplifier. The output of the high-gain fast amplifier is fed to a comparator circuit which compares the amplifier output signal to a user selected discriminator setting. The comparator output is fed to coincidence select switches on the detection unit's main control panel. The resulting output from the switch is then fed to the peak detect control circuits on the logic control board as one of eight "coincidence signals".

The shaping amplifier output signal is fed to the monitor/control panel as one of eight test points and also to the peak detector circuit. The peak detector output signal is then fed to one of eight inputs to an analog multiplexer (MUX #1 in Appendix Figure A.9). As shown in Figure 4.6, the HOLD and RESET control signals for the peak detector come from the peak detect control circuits on the logic board. The optional high-gain amplifier is used only for diagnostic applications.

The components shown on the MUX and ADC block diagram, Figure 4.7, are located on the logic board. The signals from the eight peak detectors are fed to MUX #1, are serially multiplexed, fed through a common buffer amplifier to the ADC, and finally sent to the microcontroller.

The primary factors which have established the overall system gain criteria for the analog front-end electronics are:

1. The maximum integrated photon energy to be measured by the system (i.e., 32 GeV per detector per pulse).
2. The maximum signal voltage that can be linearly processed by the system electronics (i.e., +4.82 V).

4.1.3.1 Preamplifier The deposition of photon energy in the BGO crystal produces a scintillation light pulse. The intensity of the scintillation light pulse is proportional to the amount of energy deposited. That light pulse activates the photocathode in the PMT which produces an amount of charge proportional to the intensity of the light pulse. That charge is amplified by the dynode string in the PMT. The amount of amplification (gain) is set by the high voltage applied to the PMT. The amplified charge from the anode of the PMT is fed to the preamplifier. The charge-sensitive preamplifier type that develops an output voltage pulse with a short rise-time (determined by the charge collection time) and a long exponential decay tail. The preamplifier is an inverting, negative feedback operational amplifier with a parallel resistor/capacitor (RC) feedback circuit. The collected charge is deposited on the feedback capacitor, and subsequently discharged through the resistor. The RC time constant is intentionally set to be large (on the order of 500 μ s) to minimize discharge during the charge collection time. The large time constant results in the long exponential decay tail on the preamplifier output pulse.

The conversion gain of the preamplifier is specified in volts of output signal per unit charge deposited on the input capacitor (e.g., volts per coulomb, millivolts per picocoulomb, etc.) and is approximately equal to the inverse of the feedback capacitance. The preamplifier

has a feedback capacitor of 100 pf; consequently, its conversion gain is approximately 10 Mv/Pcoul.

The maximum possible output voltage of the preamplifier is approximately +10 V. However, the maximum useable voltage from the preamplifier is +8.46 V (see Section 4.1.3.5). Therefore, the maximum useable charge at the preamplifier input is approximately 850 pcoul.

The system gain corresponding to the maximum energy for a particular sensitivity range is set by adjusting the high voltage bias on each PMT so that a maximum charge out of the PMT is 850 pcoul at the specified maximum energy. Because the BGO/PMT sensitivity (in picocoulombs per million electron volts) and the preamplifier conversion gain (in millivolts per picocoulomb) are constants independent of the photon energy, the high voltage can be set using a radioactive source having a photopeak energy less than the maximum energy. For example, in the 0-250-MeV sensitivity range, a Cs-137 source with a photopeak energy of 0.662 MeV will result in an output voltage pulse from the preamplifier of $(0.662 \text{ MeV}/250 \text{ MeV}) \times 8460 \text{ mV} = 22.4 \text{ mV}$.

4.1.3.2 Shaping Amplifier A shaping amplifier is used following the preamplifier to shape each pulse in order to obtain an accurate determination of the peak amplitude in the subsequent pulse analysis circuitry and to remove the long tail on each pulse coming from the preamplifier. The fast rise-time, long-tail pulse from the preamplifier is generated to assure complete charge collection from the PMT. However with this pulse shape, the pulse peak amplitude cannot be obtained accurately and there is a potential problem of "pulse pileup" at high pulse rates. The key characteristic of the preamplifier pulse is the peak

amplitude which is proportional to the total energy deposited in the BGO crystal. The rise-time of the pulse is too fast and the time at the peak is too short for this pulse to be adequately analyzed by the peak-height analysis circuit. For example, the peak detector circuit is not able to accurately find the peak amplitude of the pulse. Pulse pileup occurs when a second and successive pulses occur before the first pulse has returned to baseline. As a result the output pulse will be a combination of the first pulse and subsequent pulses on the tail of it. The resulting pulse height will be in error and the magnitude of the error will depend on the pulse rate and the characteristic pulse length.

The long tail on the preamplifier output pulse is reduced by differentiating this pulse in the first stage of the shaping amplifier. The time constant of the differentiator is selected to retain the fast leading edge and amplitude of the pulse and differentiate, thereby shortening, the tail. In the second stage of the shaping amplifier, the pulse is integrated to stretch the leading edge of the pulse and retain the shortened tail. The result is a shaped pulse that rises to a rounded peak and then decays fairly rapidly to zero. The time to peak is determined by the integrator time constant and is selected to be optimally compatible with peak analysis circuits. The time to peak is set at 10 μs . This is the optimum peaking time for the peak detector (see Section 4.1.3.5).

Because the pulse shape from the preamplifier is not rectangular but has a finite decay time, there will be an "undershoot" at the tail of the pulse at the differentiator/integrator output in the shaping amplifier. From transfer function analysis of the differentiator/integrator circuit, the "undershoot" occurs because of a double "pole" in the transfer function. The undershoot is eliminated by adding a "zero" to the transfer function to eliminate one pole. The technique is aptly termed "pole-zero cancellation". The zero is added by installing a resistor in parallel with the differentiating capacitor.

The following discussion of the shaping amplifier describes the Analog Front-End Schematic in Figure A.7 of Appendix A. The differentiating capacitor, C6, is 1000 pf. The pole-zero cancellation resistors are the 10-k Ω variable resistor, R3, in parallel with the 200-k Ω resistor, R4. The variable resistor allows adjustment of the pole-zero cancellation. The integrating function is achieved with two amplifiers following the differentiating capacitor. Each amplifier has a shaping time of 5 μ s determined by capacitor C8, and resistors R6 and R7 for the first amplifier and C10, R11, and R12 for the second. The two amplifiers in series result in a net shaping time of 10 μ s. The gains are approximately -0.77 and -1.31, for the first and second amplifiers, respectively. The result is a net gain of +1.0. Ideally, the total charge from the PMT is deposited on the preamplifier feedback capacitor, C5. However, some charge is lost through feedback resistors R1 and R2. This results in a signal that is slightly less in amplitude than desired. The "peaking" capacitors, C7 and C9, in the shaping amplifiers are used to recover the gain lost in the preamplifier.

4.1.3.3 Trigger Circuit The trigger circuit in the MARK-1 detection unit is designed to eliminate spurious counts from background radiation. This is accomplished by requiring coincidence counts in two or more of the detectors. Under normal single photon counting, the system will analyze any photon detected in the eight detectors whose pulse amplitude exceeds the voltage discriminator setting. The discriminator is used to eliminate false counts due to electronic noise. Under normal operating conditions the "gamma flash" photons from a pulsed radiation source are the photons of interest. The intensity of this radiation is generally sufficient to cause simultaneous pulses in two or more detectors, whereas background radiation and check sources do not normally cause simultaneous pulses in two detectors. Therefore, a trigger circuit based on coincidence counting in two or more detectors allows for the detection of real pulsed source radiation and eliminates the spurious counts due to background, etc. The trigger circuit is designed to detect a coincidence signal

in two or more detectors. If the system is going to analyze the detector pulses, the coincidence signal must be generated in the time interval prior to the peak amplitudes reaching the shaping amplifiers. This requires a separate set of fast-amplifiers in the trigger circuit. These amplifiers must be able to respond in times shorter than the rise-times of the shaping amplifiers in the peak analyzer circuits.

The drawing for the trigger circuit is shown on the Analog Front-End Schematic in Figure A.7 of Appendix A. This figure is used in the following discussions. The circuit operates using the leading edge of the signal output from the preamplifier. The preamp output is differentiated by the CR circuit, (C11 and R16). The differentiation produces a fast rise-time pulse. The differentiated signal is fed through a high gain, broad-band, two-stage amplifier (gain = 100) to increase the leading edge amplitude of low level signals of interest. The amplifier output is differentiated in CR circuit, (C13 and R24), and input to a comparator in the peak detector integrated circuit. The discriminator level of the comparator is adjusted using the 20-k Ω variable resistor, R26. The output signal is a TTL (Transistor Transistor Logic) level pulse available at pin 12 of the peak detector chip. Resistors R29 and R30 establish the TTL logic levels at approximately 0 and +5 V. The comparator outputs from eight detectors are fed to coincidence selection switches at the monitor/control panel. These switches are used to select the specific detectors used to generate a coincidence signal. The resulting coincidence signal triggers the system to perform pulse height analysis on the shaping amplifier output pulses and to store the results on the removable memory card.

4.1.3.4 Optional High Gain Amplifier A three-stage, non-inverting amplifier (gain = 200) is included in the front-end amplifier box for each detector. This amplifier is not in the circuit during normal operations, but is used primarily as an aid in troubleshooting. At some points in the circuit, the low level signals are not large enough to be effectively displayed on

an oscilloscope. In these situations, the high gain amplifier is used to amplify these voltage levels so that they can be displayed. This amplifier is inserted in the circuit between the preamplifier output and shaping amplifier input using the jumper blocks shown on the schematic, Appendix Figure A.7, between the preamplifier and fast trigger amplifier.

4.1.3.5 Peak Detector A peak detector circuit is used to detect and hold the peak value of the output pulse (a maximum of 8.46 V) from each shaping amplifier. The peak detect and hold circuit is a single integrated circuit and is shown in the Analog Front-End Schematic of Appendix A, Figure A.7. In this circuit, a capacitor is charged to the same voltage level as the peak value of the output pulse from the shaping amplifier. The capacitor is then isolated to hold the charge until the voltage can be read and processed by the on-board microcontroller.

The peak detector circuit is configured as a non-inverting operational amplifier with a gain of +1.0. Internally, there are a non-inverting input signal amplifier, a non-inverting output signal amplifier, and an inverting reset control amplifier. The holding capacitor, (C15), is connected to the output of the input signal amplifier. Control of the peak detector is via the HOLD and RESET signals that operate the switches, SW1 and SW2.

The output signal from the shaping amplifier is fed to the input peak detect circuit. This charges the holding capacitor to the peak value of the input signal. At peak time (10 μ s), SW1 is opened which isolates the capacitor. The output amplifier has a high input impedance to minimize discharge of the holding capacitor through the output amplifier. The output signal of the OP AMP, PIN 3, is fed to the multiplexer on the logic board and is also fed back to PIN 5 to achieve a gain of +1.0. The HOLD signal to SW1 is generated in the coincidence logic circuit on the logic board. The coincidence signals from the trigger circuit

activate the system to read the signals from eight peak detector circuits. After the signals are read, the coincidence logic generates the RESET signal fed to SW2. When SW2 closes, the holding capacitor is discharged (RESET) to zero back through the output circuit of the RESET control amplifier. The circuit is held in the RESET mode until the next coincidence signal occurs. The coincidence signal switches both SW1 and SW2 to the peak detect mode.

4.1.3.6 Multiplexer, Buffer Amplifier, Analog-to-Digital Converter The output signals from eight peak detector circuits are fed to an analog multiplexer on the logic board. The multiplexers, buffer amplifier, and analog-to-digital converter (ADC) are included on the schematic diagram for the Logic Board, Page 1 of 4, Appendix A, Figure A.9.

Two eight-channel multiplexer (MUX) chips (type DG538) are used in the MARK-1 detection unit. Their effective gain is +1.0. The first MUX receives the peak detector signals from eight amplifier boxes. The second MUX receives analog signals from two temperature sensors and the high voltage readout from the high voltage module. The data outputs of the MUX's are fed to a common analog buffer amplifier. The MUX's are controlled by output signals from the on-board microcontroller which will serially read the multiplexer starting with MUX 1 (inputs 0-7) and then MUX 2 (inputs 0-2). The A19 signal (to MUX PIN 17) determines which MUX chip to enable (A19 = 0 selects MUX 1; A19 = 1 selects MUX 2). The S0, S1, and S2 signals (MUX PINS 16, 15, 14, respectively) are used as address lines to the MUX to select 1 of 8 input signals. The WR signal, MUX PIN 13, latches the selected address in the MUX and feeds the selected input to the buffer amplifier. The buffer amplifier is a noninverting operational amplifier whose output is fed to the analog input of the ADC (AIN - PIN 26). The ADC is a 12-bit type CS5012 ADC. The ADC LATCH signal (ADC PIN 1) from the microcontroller latches the input analog signal on an internal capacitor in the ADC. At the end of the ADC conversion,

the ADCRDY signal from PIN 38 of the ADC is fed to PIN 14 of the microcontroller. The microcontroller then generates the EN and RD signals to PINS 21 and 22 of the ADC to read and store the digital data from PINS 2-9 of the ADC. Because the output data from the ADC is only 8 bits wide, two READS are required to read one data conversion. The ADC will output the eight high-order bits on the first READ, and then the four low-order bits with four trailing zeroes on the second READ.

The gains of the MUX, AMP, ADC circuits are determined by the operating characteristics and requirements of the various components throughout the analog system. As mentioned in Section 4.1.3.5, the peak detector is limited to a maximum linear input of +8.46 V and, with a gain of +1.0, the peak detector output is also limited to +8.46 V. The maximum input to the ADC must be below the ADC supply voltage of +5 Vdc. This maximum is set by a reference voltage applied to PIN 28 of the ADC. This reference voltage is generated by a voltage regulator, type AD586, and is set at +4.82 V. Therefore, the input signals to MUX #1 must be reduced from the maximum of +8.46 V from the peak detector to a maximum of +4.82 V at the ADC input. This voltage reduction applies only to the radiation detector signals being input to MUX #1 and is accomplished using a resistor voltage divider. The 10-k Ω and 39.2-k Ω resistors at the inputs of MUX #1, along with the two 10-k Ω resistors at the input of the buffer amplifier produce the design gain of 0.56974.

Two temperature sensors, type LM135, are used in the MARK-1 detection unit. They are powered from constant current sources (type LM134). The output voltages from these sensors are generated at a rate of 10 mV per degree Kelvin, and will operate over a temperature range of -50°C to +150°C (223°K to 423°K). The corresponding output voltages for this temperature range are +2.23 V to +4.23 V.

The high voltage reading is obtained from a 1:1000 voltage divider in the high voltage module (see Section 4.1.1.3). Typical operating voltages in the detection unit range from about 350 V to 750 V resulting in a high voltage reading of 350 mV to 750 mV. These values are within the +4.82 V maximum for the ADC. The buffer amplifier is designed to accept signals from both MUX #1 and MUX #2 and has a gain of +1.0 to accommodate the temperature and high voltage signals.

4.1.4 Logic Circuits

Internal control of the detection unit is accomplished using an 87C51FA microcontroller. The logic circuits include not only the microcontroller but also peripheral support circuits and special purpose, digital circuits. All of the logic circuits are located on the logic board located on the left side of the detection unit suitcase (see Figure 2.2).

The support circuits include a real-time-clock (RTC) integrated circuit, a random access memory (RAM) card, a function decoder circuit, and an RS232 serial input/output (I/O) driver. The special purpose circuits are: the digital circuits required for automatic gain control, the logic circuits used for control of the analog peak detectors, and the coincidence logic circuits.

4.1.4.1 Microcontroller The microcontroller unit used is an 87C51FA, a member of the INTEL 8051 family. Characteristics of special interest for the MARK-1 application are:

1. 8 kbytes of on-board electrically programmable and erasable, programmable, read-only memory (EPROM) for program memory.

2. 256 bytes of on-board random access memory (RAM) for temporary data storage.
3. 32 programmable I/O lines. All 32 lines are individually and independently programmable. These lines are logically organized as 4 each 8-bit input/output (I/O) ports (#0-#3). In addition to their use as general purpose I/O ports, they are also used for special purpose programmable functions. Port #0 is used as the multiplexed low-order address byte and data bus during external memory addressing (see #4 below). Port #1 can be used (but is not) for special internal timer control. Port #1 is used to output 4 higher-order address bits (#16-#19), 3 function select bits (S0-S2), and an ENABLE signal. Port #2 is used to output the high-order address byte (A8-A15) during external memory addressing. Port #3 is used for the serial port I/O lines (RXD and TXD), to output the read and write control pulses (RD and WR), receive external interrupts (EVENT and ADCRDY), output the ADC LATCH control, and output the high-order address byte (A20).
4. 64 kbytes of external memory addressing capability. For this application, the low-order address byte is time-multiplexed with data on I/O port #0 and the high-order address byte is output on port #2. In the MARK-1, external memory addressing capability is extended to 1 Mbyte by outputting an additional four higher-order bits on port #1 (A16-A19). External memory may be either program or data memory. The actual external memory of the MARK-1 is a 512-kbyte data memory card.
5. On-board serial interface used as an RS232 serial I/O port. The interface includes send and receive data latches. Data format and baud rate are programmable. The interface also includes a serial port interrupt capability.

6. The on-board oscillator to control the 87C51FA is triggered from an external quartz crystal. The crystal frequency is 11.0592 MHz. The external circuit containing the crystal is designed to pick off the oscillator frequency and feed the output to the programmable gate array, PGA1, through a 2N3904 NPN transistor (see Section 4.1.4.5).

4.1.4.2 Real-Time-Clock (RTC) - The block diagram for the RTC was shown in Figure 4.1. In the MARK-1 it is a National Semiconductor type MM58167A microprocessor real-time-clock. This device is triggered from a 32.768 kHz quartz crystal to generate month (01-12), day of month (01-31), hours (00-23), minutes (00-59), seconds (00-59), and milliseconds (000-999). The data to/from the RTC is via an 8-bit clock data bus (CD0-CD7) and is in binary coded decimal (BCD) format. The RTC can be preset from an external laptop computer through the serial I/O port. An on-board software routine directs the operator of the laptop computer to successively input the month, day of month, hour, and minute to the MARK-1. The operator inputs requested data as two decimal digits. For example, the third month would be entered as 03 (not 3 alone). The input is converted to two BCD digits (e.g., 03 converts to 0000 0011) and is loaded to the RTC. The final entry is the minutes. Once the minutes have been entered, the control program loads the RTC, presets the seconds and milliseconds to zero, and starts the RTC running. Selection of the time function (e.g., month, day, hour, etc.) is done by address lines A16-A20 from the on-board microcontroller. The RTC decodes the lines to specify the time function. The time function decode applies to both the write to and read from the RTC. The WR and RD lines from the microcontroller select write or read, respectively. The RTC is addressed from any other support or special purpose circuit by the chip select lines SEL2 and SEL3. These signals are generated by the function select circuit in a programmable gate array (PGA2) (see 4.1.4.3 below). In the same manner that any WRITE to the RTC consists of two BCD

digits, all READS are two BCD digits. The MARK-1 reads out the month, day, hour, minutes, seconds, and milliseconds. Each of these reads is two BCD digits except milliseconds, which requires two reads of two BCD digits each. For example, 249 ms is read in two steps: 24 and 90 where the trailing digit is always 0. The block diagram for the RTC shows two sets of parallel latches on the data lines between the RTC and the microcontroller. These latches are used to isolate the 8-bit clock data bus from the 8-bit microcontroller data bus. For a write to the RTC, data on the microcontroller data bus are loaded in the input latches, then transferred to the RTC on the clock data bus. Conversely, for a read from the RTC, the clock bus data is loaded to the output latches, then transferred to the microcontroller on the microcontroller data bus. The latches are implemented in a programmable gate array (PGA2). (See the schematic diagram Logic Board, Page 2 of 4, Figure A.10 of Appendix A.)

When system power is turned OFF, the RTC receives its power from a +4.5-Vdc backup battery circuit shown in Figure 4.1.

4.1.4.3 Function Decoder Various support and special purpose functions are controlled by the microcontroller. A maximum of 8 functions can be controlled by three input lines: S0-S2 (see bottom of Figure A.11 in Appendix A). The decoder for these input lines is implemented in the programmable gate array #2 (PGA2). The binary value (0-7) of the S0-S2 lines selects 1 of 8 possible output lines (0-7). The selected output line goes to a logic LO during the time the enable pulse, EN, is received from the microcontroller.

In the MARK-1 detection unit, five of eight possible functions can be selected (i.e., SEL1, SEL2, SEL3, SEL5, and SEL6). The SEL1 function select line is fed to the RAM memory card to enable the card to write data from the microcontroller data bus to the card.

The SEL2 and SEL3 lines are used to control the flow of data between the microcontroller and real-time-clock through the input and output RTC data latches (see Section 4.1.4.2 above). The SEL5 and SEL6 function select lines are used to control the logic circuits in the gain control system.

The input and output RTC data latches are implemented in PGA2. For the WRITE to the RTC from the microcontroller, SEL3 is first ANDed with WR to load microcontroller data from the microcontroller data bus (D0-D7) to the input latches. Then SEL2 is ANDed with WR to transfer data from the input latches to the RTC data bus (CD0-CD7). The SEL2 signal is also fed to the RTC chip as a "CHIP SELECT" signal. The WR signal to the RTC chip causes the generation of the RDY signal internally in the RTC chip. When this chip determines that valid data from the RTC data bus has been latched into the RTC, the RDY signal is released.

For a READ from the RTC to the microcontroller, the SEL2 (CHIP SELECT) along with the RD signal to the RTC outputs data from the RTC to the RTC data bus. The RDY signal is generated within the RTC chip and is fed to PGA2. When the RTC determines that valid output data is on the RTC data bus, then RDY is released. The trailing edge of RDY loads data from the RTC data bus to the output latches. Then SEL3 is ANDed with RD to transfer data from the output latches to the microcontroller data bus (D0-D7).

4.1.4.4 Gain Control System Control from the on-board microcontroller permits the MARK-1 detection unit to alter the detector gain of the system during operation. The microcontroller determines the detector gain required to ensure that all detector responses are within acceptable limits (i.e., no "underflows" and no "overflows" in any of the eight detector responses) and automatically changes the output high voltage to achieve the required

gain by changing the high voltage control word to the DAC. These voltage changes are recorded in the output data.

The microcontroller is programmed to generate any one of eight high voltage control words corresponding to the eight different sensitivity ranges available to the system. These control voltages are fixed and unique for each MARK-1 unit. This uniqueness occurs because the detectors used in one unit are not necessarily matched with the detectors in other units (see Section 5 of Volume 2).

The logic circuits used in the gain control system consist of a digital-to-analog convertor (DAC) with a buffer amplifier to control the high voltage module output (see Figure A.10 in Appendix A) and an 8-bit gain data latch with gates to store gain information during system power OFF (see Figure A.9 of Appendix A).

Whenever a gain step is either manually selected using the external laptop computer or automatically changed by the on-board microcontroller, gain information consisting of the gain step and control word are loaded into the latch. When system power is turned OFF the data in the latch are retained using a +4.5-V backup battery. When system power is turned back ON and the system is in the manual gain mode, the gain data in the latch are read and the gain setting is returned to the value that existed at the time the power was turned OFF. When the power is turned back ON and the system is in the automatic gain mode, the gain data in the latch are reset to the highest system gain setting and the system resumes operation at the highest gain setting.

The gain data latch is a type 74HC573 eight bit latch. This is a tri-state device controlled by the latch enable (LE) and output enable (OE) inputs. In the MARK-1 detection

unit, the input and output lines of each bit are tied together and to the respective lines on the microcontroller data bus (D0-D7). The latch is selected for either a READ or WRITE by the function select line SEL5. The SEL5, WR, and RD lines are fed to an open collector OR gate as normally HI (i.e., High Logic Level) lines. The resulting OR gate outputs, Y2 and Y4, are normally HI. The Y4 output is fed directly to the OE input of the latch. With OE=HI, the Q outputs of the latch are in a high impedance state and have no effect on the microcontroller data bus lines, and any data on this data bus will have no effect on the latch. To load data to the latch, the SEL5 and WR lines are both taken LO (i.e., Low Logic Level). The Y2 OR gate output goes LO, is inverted by the 74HCO4 inverter to a HI, and is fed to the LE input of the latch. Data from the microcontroller are placed on the D0-D7 bus lines to the latch inputs. When either SEL5 or WR is removed (goes HI), the LE input to the latch goes LO and locks data into the latch. Note that OE is still HI and the latch Q outputs are in a high impedance state.

To READ the latch, the SEL5 and RD lines are both taken LO. The Y4 output of the OR gate goes LO and is fed to the OE input of the latch. This enables the Q outputs and applies the latch contents to the microcontroller data bus lines (D0-D7). The microcontroller then reads the bus lines.

When system power is turned OFF, the two 10-k Ω pullup resistors between the Y2 and Y4 OR gate outputs and the backup battery hold the OE and LE inputs to the latch to HI and LO, respectively. The backup battery retains data in the latch. The HI level to OE keeps the Q outputs in a high impedance state to prevent latch data from adversely affecting the microcontroller data bus lines at power ON. The LE input is kept LO at power ON to prevent the possible HI to LO transition from the OR gate output through the inverter from loading arbitrary data from the microcontroller data bus.

The output voltage from the high voltage module is controlled by a low level analog control voltage. The control voltage is generated by a digital-to-analog converter (DAC), Type AD557, and buffer amplifier on the logic board. (See the schematic diagram in Figure A.10 of Appendix A.)

Whenever a gain change is either manually selected by the operator or automatically changed by the on-board microcontroller, the microcontroller generates an 8-bit digital control word that is proportional to the required high voltage. The control word is fed out on the microcontroller data bus (D0-D7) to the DAC which has a built-in 8-bit latch. The function code select circuit (see 4.1.4.3 above) generates the SEL6 function select line. The SEL6 signal is ANDed with WR, and feeds the resultant WRSEL6 signal out to the DAC. This signal loads the digital control word to the internal latch and feeds the internal latch output to the DAC circuit. The DAC output is fed to the buffer amplifier, and hence, to the high voltage module. The output voltage from the DAC is limited to 0 to +2.56 V. The required control voltage to the high voltage module is -3.54 to -7.30 V. The buffer amplifier is designed to meet these input/output requirements. The LF411 amplifier is connected as an inverting operational amplifier. The 35.7-k Ω feedback resistor and 121-k Ω input resistor from the +12-Vdc power line yields a fixed DC output of -3.54 V. The feedback resistor together with the 24.3-k Ω input resistor from the DAC yields a second output of 0 to -3.76 V for a net output of -3.54 to -7.30 V.

4.1.4.5 Coincidence Logic and Peak Detector Control Figure 4.8 is a block diagram of the coincidence and peak detector control circuits. Figure 4.9 is a simplified timing diagram

BLOCK DIAGRAM - COINCIDENCE/PEAK DETECTOR CONTROL

Figure 4.8 Coincidence and Peak Detector Control Block Diagram.

SIMPLIFIED TIMING DIAGRAM
COINCIDENCE LOGIC AND PEAK DETECTOR CONTROL

Figure 4.9 Timing Diagram - Peak Detector Control.

of the control circuit. These circuits accept the outputs of the coincidence select switches at the control panel and generate:

- (1) the "EVENT" interrupt to the microcontroller to initiate data acquisition,
- (2) the control signals HOLD and RESET to the analog peak detectors, and
- (3) the "TRIG OUT" signal to the control panel as a test point.

See Section 4.1.3.5 for a description of the analog peak detectors.

These circuits are implemented in a programmable gate array (PGA) manufactured by the XILINX Corporation (Type 2064-48/50). Refer to the schematic diagram presented in Figure A.10 of Appendix A for the interconnects of the PGA to the system, and the schematic diagram shown in Figure A.12 for the internal functions of the PGA. A PGA is basically a series of independent logic blocks, each comprised of a number of logic gates and flip-flops. To implement a given function in a PGA, the interconnects within and between logic blocks are "programmed" into the device to achieve the logic function. The functional design of the PGA is accomplished on an external computer that generates the program code (configuration file) to download into the PGA. The program code can be downloaded from either the external computer directly to the PGA or can be loaded to a read-only memory (ROM) from the computer and then downloaded from the ROM to the PGA. The PGA's in the MARK-1 detection unit are downloaded from on-board ROMs.

When the main power is turned ON, the PGA generates a "D/P" signal (DONE/-PROGRAM) to enable the ROM to download the configuration file to the PGA. The PGA also generates the clock (CCLK) to transfer the file to the PGA. The file is serially transferred from the ROM to the PGA on the "DIN" (Data In) line. If the file is

downloaded from an external computer to the PGA, the ROM must be removed from the board. In this case, the computer generates the "D/P" and "CCLK" signals and transfers data on the "DIN" line. The mode select lines (M0-M2) are used to select either serial transfer from the computer (M0-M2 are tied to +5 Vdc) or from the on-board ROM (M0-M2 are tied to ground). Pins are available on the logic board to tie to these various points. A special purpose cable is required from the external computer to the logic board. The RESET switch to pin #31 of the PGA can be used to reset all flip-flops in the PGA. This is normally required prior to a download from an external computer, but not from the ROM. The reset will not affect the PGA configuration.

The operation of the coincidence logic and peak detector control circuits is initiated by the signals from the coincidence select switches at the monitor/control panel (see Sections 4.1.3.3 and 4.1.3.5). The coincidence circuit is shown at the upper left of the Figure A.12 schematic diagram. This circuit uses an 8-input AND gate that requires eight HI signals from the select switches to generate the COINC signal. Any switch that is not selected feeds a logic HI from the +5-Vdc power line. A switch that is selected feeds a logic HI signal from the appropriate comparator in the analog front-end circuits when a valid gamma pulse is detected.

The remainder of the circuits in the PGA are designed to generate the TRIG OUT, EVENT, HOLD, and RESET signals. The operation of the circuit is summarized below:

1. A coincidence at the inputs generates TGSW
2. TGSW generates HOLD = LO and RESET = LO. This is the PEAK DETECT mode at the peak detectors. The test point, TRIG OUT, is also generated.

3. Approximately 10.85 μs after TGSW, TRIG OUT is reset, HOLD goes HI, and EVENT = LO is generated. RESET = LO is unchanged. The combination of HOLD = HI and RESET = LO is the requirement for the peak detector HOLD mode. EVENT = LO is the interrupt signal to the microcontroller to start data acquisition. EVENT lasts for approximately 2.17 μs .
4. The peak detector HOLD mode lasts for approximately 662 μs after TGSW. During this time, the microcontroller reads and stores the peak value from eight peak detector circuits.
5. Approximately 662 μs after the coincidence (TGSW) occurs, RESET = HI is generated. With HOLD = HI and RESET = HI, the peak detectors are in the RESET mode.
6. The circuit remains in the RESET mode until the next coincidence.

The following detailed description utilizes Figure 4.9 and the schematic diagram presented as Figure A.12 in Appendix A.

The COINC signal with RSTBR (initially HI) generates TGSW at the output of an AND gate. The TGSW signal is fed to the HOLD flip-flop (lower left of schematic) to generate HOLD. HOLD is buffered and fed to PIN 41 of the PGA as the "TRIG OUT" signal. "TRIG OUT" is a test point at the monitor panel to signify that a coincidence has occurred. HOLD is inverted (HLDBR) and fed out to PIN 9 as the "HOLD" = LO signal to eight peak detectors. HOLD is also used to set the RST flip-flop (bottom of schematic) which is inverted (RSTBR = LO), buffered, and fed out to PIN 8 of the PGA as "RESET" = LO to

eight peak detectors. "HOLD" = "RESET" = LO is the combination required for the peak detect mode at the peak detectors. RSTBR = LO is used to enable the RSTTC counter and FINCT flip-flop (bottom of schematic), enable the LTC and HLDIRST counters, and to disable the TGSW AND gate to prevent any other COINC signals from triggering the system while this COINC is being processed.

TGSW is also fed directly to the clock input of the HLDENPE flip-flop (lower right of schematic) and the clock input of the ENPE flip-flop (upper right of schematic) through an OR gate to set the flip-flops. The EVENTBR signal to the D inputs of the flop-flops is initially HI. The HLDENPE and ENPE signals are fed to the HLDIRST and LTC counters respectively as preset enable (PE) signals.

The input clock at PIN 47 of the PGA (left center of schematic) is fed to the PGA from the microcontroller (see Section 4.1.4.1). The SYSCLK flip-flop divides the input frequency by 2 to obtain SYSCLK with a frequency of 5.5296 MHz (period ≈ 180.845 ns). The next SYSCLK pulse (positive-going) after the preset enable signals are applied will preload the LTC and HLDIRST counters with initial values selected by external delay switches DLY0-DLY5. The LTC counter is clocked directly with SYSCLK. The HLDIRST counter is clocked through the PSECLK AND gate and the ENLTC OR gate.

The ENPE and HLDENPE signals are fed through an OR gate to enable the CLRPE counter (upper right of schematic). This is a divide-by-2 counter clocked with SYSCLK ANDed with EVENTBR (initially HI). The result of the counter is a delay of CLRPE at the terminal count (TC) output of the counter of 2 times SYSCLK period ≈ 362 ns after TGSW. The delay is generated to allow time for the LTC and HLDIRST counters to be preset. The CLRPE signal clears the ENPE and HLDENPE flip-flops and sets the LCPE flip-flop (lower

right of schematic). The LCPE signal is fed to the LTC and HLDRST counters as count enable (CE) signals. LCPE also clears the CLRPE counter. Note that the CLRPE signal also resets the ENPE flip-flop.

At this point (≈ 362 ns after TGSW), the "TRIG OUT" signal is HI (a coincidence has occurred), the "HOLD" and "RESET" signals to the peak detectors are LO and the LTC and HLDRST counters have been preset and enabled in the count mode with LCPE. LCPE will remain HI until the LCPE flip-flop is reset when RSTBR goes HI. Any other CCINC signals are locked out until RSTBR goes HI. RSTBR will remain LO until the microcontroller reads and stores data from eight peak detectors.

The LTC counter is preset with a value of three. This is a 4-bit up counter that outputs a terminal count (LTC) pulse at the end of the maximum count of fifteen. This counter is then a divide-by-13 counter which, in effect, multiplies the input clock period by 13. The LTC pulse is fed to an OR gate that sets the ENPE flip-flop (top center of Figure A.7). ENPE is the preset enable for the LTC counter. The next SYSCLK presets the counter. Counting is delayed until the CLRPE counter generates CLRPE to clear the ENPE flip-flop. This amounts to a delay of approximately 362 ns after preset before the LTC counter can start counting. The LTC counter requires about 13 times SYSCLK period $\approx 2.351 \mu\text{s}$ to generate each LTC. Then the time from presetting the LTC counter to the generation of the LTC pulse is equal to the CLRPE delay plus the count up time of the LTC counter. This is approximately 362 ns plus $2.351 \mu\text{s} \approx 2.713 \mu\text{s}$. The LTC counter will generate LTC pulses at this rate as long as the count enable input (LCPE) is applied to the counter.

The LTC pulses are fed through the ENLTC OR gate to clock the HLDRST counter. The HLDRST counter is initially preset as a divide-by-4 counter. Then 4 LTC clock pulses

are required to obtain the first HLDRST pulse. This is $4 \times 2.713 \mu\text{s} \approx 10.85 \mu\text{s}$ after TGSW. This slightly exceeds the time-to-peak of $10 \mu\text{s}$ for the shaping amplifiers and guarantees the amplifiers will peak before HLDRST. The HLDRST pulse is fed to the reset input of the HOLD flip-flop (lower left, Figure A.7). When HOLD goes LO, the TRIG OUT signal at PIN 41 goes LO. HOLD is inverted (HLDBR = HI) and drives the HOLD output at PIN 9 HI. The RESET output at PIN 8 remains LO. The combination if RESET = LO and HOLD = HI is the combination required for the hold mode at the peak detectors.

The inverted output of the HOLD flip-flop (HLDBR) going HI sets the EVENT flip-flop. EVENT is inverted (EVENTBR), buffered, and fed out at PIN 48 of the PGA as the negative-going interrupt pulse (EVENT) to the microcontroller. This pulse starts the acquisition of data by the microcontroller. EVENT is fed to the count enable input of a divide-by-12 counter whose terminal count clears the EVENT flip-flop. This makes the EVENT pulse approximately $12 \times \text{SYSCLK period} \approx 2.17 \mu\text{s}$ wide. This is sufficiently long to be recognized by the microcontroller as a valid interrupt pulse.

Note, that the TGSW signal is generated only once as result of a coincidence, COINC (see upper left of schematic). The RSTBR signal disables the TGSW AND gate after the initial COINC and until the RST flip-flop (lower right) is cleared.

The initial HLDRST signal was generated at the output of a counter that had been preset to a divide-by-4 by the preset enable signal, HLDENPE. HLDENPE is generated only once during the processing of a COINC because TGSW occurs only once. After the initial HLDRST pulse at $10.85 \mu\text{s}$ after TGSW, the counter will no longer be preset and will

convert to a divide-by-16 counter. Subsequent HLDRST pulses will then occur every 16 times LTC period = $16 \times 2.713 \mu\text{s} \approx 43.4 \mu\text{s}$.

The HLDRST pulses are fed to a divide-by-16 counter, RSTTC (bottom of schematic). The initial HLDRST clocks the counter to a count of 1 at $10.85 \mu\text{s}$ after TGSW. Thereafter, the counter effectively becomes a divide-by-15 counter clocked at a rate of $43.4 \mu\text{s}$, such that the terminal count, RSTTC, will occur at approximately: $(10.85 \mu\text{s} + (15 \times 43.4 \mu\text{s})) \approx 662 \mu\text{s}$ after TGSW. This guarantees that the microcontroller will have adequate time to read peak data from all eight peak detectors.

The RSTTC output sets the FINCT flip-flop, which in turn resets the RST flip-flop. RST is inverted, buffered, and fed out to PGA PIN 8. This is the positive going trailing edge of RESET to the peak detectors. With HOLD (PGA PIN 9) previously set HI at $10.85 \mu\text{s}$, and RESET now going HI, this is the combination required for the reset mode at the peak detectors. The inverted RST signal (RSTBR = HI) resets the RSTTC, FINCT, LCPE flip-flops, and the LTC and HLDRST counters, and enables the TGSW AND gate. The circuit is now in a condition to receive the next coincidence pulse. The procedure is repeated for each coincidence.

4.1.4.6 Serial I/O Communication between the MARK-1 system and the external laptop computer is via an RS232 serial I/O port. The primary need for the computer/MARK-1 communication link is to initialize the MARK-1 detection unit. Initialization consists of setting the real-time-clock, and selecting the initial gain control parameters. These parameters are: (1) mode selection (automatic gain control or fixed manual control) and (2) the initial gain setting. The on-board microcontroller generates prompts that are transmitted via the serial port and displayed on the external computer

screen. The operator responses are entered through the external computer and transferred to the MARK-1 unit via the serial port to the microcontroller.

Acquisition data and acquisition program sequence indicators can also be transmitted from the microcontroller to the external computer via the serial port as the data is being accumulated. Since the serial port communication link slows down the acquisition data rate, the computer is not normally connected to the detector unit during actual data acquisition. This link is used primarily for initialization and troubleshooting purposes. The acquisition data are: (1) date/time of each detected radiation pulse, (2) eight detector peak values for each pulse, (3) two temperature values, (4) the gain mode, (5) the gain change indicator, (6) the detector high voltage, (7) gain step, and (8) the gain control word. These values are displayed in a data block on the computer screen for each radiation pulse. The data block appears on the computer display in a sequential order as the data are being acquired. A number of program sequence indicators are also displayed. These can be used to indicate the point of operation within the data acquisition process.

The serial I/O circuits consist of two sections. The serial I/O port proper is built into the on-board microcontroller. The receive and transmit registers and the conversion of parallel-to-serial and serial-to-parallel data is accomplished in the microcontroller. Control of the baud rate and number of bits is programmable. The receive and transmit lines at the microcontroller are at CMOS logic levels. The second section of the serial I/O circuit is a type LT1080 driver/receiver chip from Linear Technology. The LT1080 converts CMOS logic levels on the receive and transmit lines to RS232 voltage levels of +9.0 V and -8.5 V.

The MARK-1 RS232 parameters are programmed using a transparent subroutine in the on-board microcontroller. The subroutine sets the baud rate to 19.2 kbaud/s, and the bit stream to 1 start bit, 8 data bits, no parity, and 1 stop bit.

4.1.4.7 Memory Card The logic board containing the microcontroller and support peripherals is a special purpose computer. The memory section of this "computer" is divided into three basic sections. The first is the program memory. This is an 8-kbyte erasable, programmable, read-only memory (EPROM) contained within the microcontroller. The second is a 256-byte random access memory (RAM) contained within the microcontroller and used as temporary data storage. The third memory section is a 512-kbyte static RAM (SRAM) used for "permanent" data storage. This memory is external to the microcontroller. It is contained in a credit-card size plug-in module and is referred to as the memory card. The SRAM in the MARK-1 detection unit is a model RBC512IE10 manufactured by Epson America. Data transfer into/out of this device is via a 40-pin connector. It has a built-in exchangeable battery which retains power and data when the card is removed from the system. The card has a WRITE PROTECT switch to protect recorded data.

Addressing the 512 kbytes in the SRAM requires 19 address lines. The microcontroller has direct addressing capability of 64 kbytes of external memory, which requires 16 address lines (A0-A15). Four additional address lines (A16-A19) are output on the microcontroller port #1 (pins #1-4). The memory card uses only 3 of these 4 lines to address 512 kbytes. Address lines A8-A15 are output on microcontroller port 2. The low order address lines, A0-A7, are automatically time-multiplexed with 8 data bits on microcontroller port #0. (See schematic diagrams Figure A.9 through A.11 in Appendix A.)

Time-multiplexing is accomplished by first outputting 8 address bits (A0-A7) on microcontroller port #0, outputting the control line, ALE, (microcontroller pin #30) to load the address bits to an external latch, then outputting data bits, D0-D7, on microcontroller port #0. This now makes the following available to the memory card:

1. 8 data bits (D0-D7) from microcontroller port #0,
2. 8 low-order address bits (A0-A7) from an external latch,
3. 8 higher-order address bits (A8-A15) from microcontroller port #2,
4. 4 additional address bits (A16-A19) from microcontroller port #1.

In addition, the S0-S2 function select lines and the ENABLE pulse from the microcontroller are fed out on port #1 simultaneously with address bits A16-A19.

The external latch to hold address bits A0-A7 and the function select decoder are implemented in a programmable gate array (PGA2) (see schematic diagram Figure A.11). The decoded function select line, SEL1, is fed to the memory card as the chip enable input. The address bits are fed to the PGA on the input data lines, D0-D7, (PGA2 pins #2-9). The control line, ALE, is the clock input to the 8-bit latch. The output of the latch is the 8 address lines, A0-A7, to the memory card. Data (D0-D7) are loaded to the memory card by the application of the write pulse (WR) from the microcontroller.

4.1.5 Operator Control

All operator controls for the MARK-1 system are contained in the monitor/control panel, the remote control box, or interface to the unit through an external computer panel. The control panel is located on the left side of the detection unit (see Figure 2.2, Figure 4.10 and Appendix A, Figure A.14).

The operator controls that require an external computer are those dealing with initialization and troubleshooting of the unit. The computer connects to the system via the RS232 serial I/O interface. The connector is a 9-PIN D-connector mounted on the monitor/control panel.

4.1.5.1 Monitor/Control Panel Figure 4.10 is a diagram of the Monitor/Control Panel and listed below are the items found on that panel.

1. Power ON/OFF switch which supplies +12 Vdc from the battery to the low voltage and high voltage modules.
2. Eight coincidence select switches. Two inputs are wired to each of eight switches. A +5-Vdc (HI) input is fed through a 10-k Ω pullup resistor and through the switch if the switch is not selected for coincidence. The output from the appropriate analog comparator is the second input to the switch and is fed through the switch if the switch is selected for coincidence. (See Sections 4.1.3.3 and 4.1.4.5.)
3. Power test switch and power ON/OFF LED indicator. An LED lamp is installed as a power ON/OFF monitor and is wired in series with a 1-k Ω current-limiting resistor

Figure 4.10 Monitor/Control Panel.

and a momentary contact push-button switch. A normally open switch is used to prevent a constant current drain on the system battery. The switch is depressed to monitor the ON/OFF state of the system power.

4. Power test points: +12V, -12V, +5V, -5V, Battery (12V), High Voltage, System ground.
5. Analog signal monitors. There are eight monitor points; one monitor for each shaping amplifier output signal (see Section 4.1.3.2).
6. Trigger monitor. This test point allows for monitoring of the signal from the COINCIDENCE/PEAK DETECTOR CONTROL circuit (see Section 4.1.4.5).
7. Battery charger connector. Charging current from an external charger unit is input through a Lemo connector mounted on the outside of the detection unit suitcase. The connector is located on the left side of the suitcase. A 1N4001 diode is wired in series with the internal connection to the positive terminal of the battery to allow in-circuit charging of the 12-V system battery.
8. RS232 Input/Output serial data port.

4.1.5.2 Remote Control Box There are two functions which can be implemented by the remote control box. The first is the remote control of the system power ON/OFF circuit and the second is remote control of the data acquisition.

The remote control relay in the control panel module is a latching type relay. Energizing that relay is accomplished by applying a momentary voltage to its coil. The polarity of the applied voltage determines whether the contacts are opened or closed. In either case, the contacts are "latched" in the selected position and power can be removed from the coil. (See the schematic diagram, "LOW VOLTAGE POWER SUPPLIES" Appendix Figure A.1, for the remote control box wiring.) The power to energize the relay is supplied from a +18-Vdc battery (two 9-V alkaline batteries). A 2-pole, 2-position switch is used to select the polarity of the voltage fed to the relay. A momentary contact, normally open, push-button switch is used to apply the battery voltage to the relay coil.

An LED lamp is installed in the remote control box as a power ON/OFF monitor. The LED lamp is wired in series with a momentary contact push-button switch. The series lamp and switch are connected in parallel with the lamp and switch in the control panel such that either lamp can be used as a power ON/OFF monitor.

To control the data acquisition without turning the system power ON/OFF, a coincidence disable switch is installed in the remote control box. In the disable position, the switch grounds the 10-k Ω pullup resistor at the eighth coincidence select switch in the monitor/control panel. The ground disables the coincidence select circuits on the logic board and creates a PAUSE in the data acquisition. When the switch is moved back to the enable position (i.e., open), data acquisition will resume without starting over and without erasing the data taken before the PAUSE was initiated. Data acquisition continues on the memory card without presetting the memory address to zero.

4.2 MARK-1 Support Unit

The support unit is contained in an aluminum suitcase that contains all necessary equipment to support the MARK-1 detection unit. This includes the portable computer, card reader, printer, cables, documentation, and miscellaneous equipment.

4.2.1 Portable Computer

The portable computer used in the MARK-1 support unit is employed for system initialization, troubleshooting, and data analysis. The computer is a battery operated, portable COMPAQ LTE-286 Model 40 laptop computer. It measures 8.5" x 11" x 1.9" and weighs approximately 6.7 lb. It is equipped with 2.688 Mbytes of RAM, a 40-Mbyte hard disk; and a 3.5" floppy disk drive. Commercial software includes DOS Version 4.01, NORTON COMMANDER, CROSSTALK, and QUICKBASIC Version 4.50. A software package installed as SA10PR from Needham Electronics operates the Prom Programmer used to read the data memory cards. A battery charger is provided to recharge the computer battery.

4.2.2 Memory Card Reader

The transfer of data from the MARK-1 detection unit data memory cards to the laptop computer for data analysis is accomplished using a Needham Electronics PROM programmer, Model SA-10. It is designed for use as either a memory chip programmer or reader, but in this application, it is used as a reader of the data memory cards. The buffer memory in the SA-10 has been expanded from the standard 128 kbytes to 512 kbytes to

accommodate data from the memory cards. A Needham Model SAMC-10 adapter is provided to insert the cards in the reader.

Reading a memory card is a two step process. Card data are first read to the buffer memory in the SA-10 and then transferred to the laptop computer. A software package installed in the laptop computer is used for the operation of the SA-10. This is a menu driven package that directs the operator for the complete data transfer. See *Volume 2: Operations Manual for the MARK-1 Pulsed Ionizing Radiation Detection System*, Section 4.3 for operation of the memory card reader.

4.2.3 Printer

A printer, Hewlett-Packard Model 2225P Thinkjet, is provided to obtain hard copies of data analysis results. Its dimensions are 8.5" x 11.0" x 3.5". It uses 8.5" x 11" bifold printer paper. It is battery powered and a battery charger is provided to recharge the battery pack.

4.2.4 Cables

All the necessary cables are carried in the MARK-1 Support Unit for system connections including initialization, remote control operation, and data card reading. Figure 4.11 shows the system connections for each of these operations.

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Component Description

SYSTEM HOOKUP FOR INITIALIZATION

SYSTEM HOOKUP FOR NORMAL OPERATION

SYSTEM HOOKUP TO READ CARDS

Figure 4.11 MARK-1 System Connections.

4.2.5 Documentation

Sufficient documentation is included with the support unit to operate the MARK-1 system during field use. The MARK-1 OPERATORS MANUAL is the primary documentation. It provides comprehensive instructions for operating the MARK-1 system and also the laptop computer and the card reader in this application. Vendor supplied operation manuals for the portable computer, the Hewlett-Packard printer, and the Needham card reader are available as supplemental information but are not included as part of the support unit.

4.2.6 Miscellaneous Supplies

The support unit should also contain enough printer paper, memory cards, floppy disks, test cables, etc. for the mission at hand. Prior to commencement of a mission the operator should inventory the items covered and replenish the stock where necessary.

5. SOFTWARE

This section describes the software involved in the development and operation of the MARK-1 system. The software development kit used to write the data acquisition software is briefly described. The acquisition and data analysis programs are described in greater detail. Additional information, consisting of flowcharts and program listings for the acquisition and analysis programs, are included in Appendix B.

5.1 Acquisition and Control Software

5.1.1 Software Development Kit

The software development kit used for the MARK-1 model 87C51FA microcontroller is the "Archimedes C-51 Cross Compiler Software Development Kit" for MCS-51 8-bit microcontrollers. The kit was developed by Archimedes Software Inc. It includes a C-compiler, assembler, linker, and librarian. The object codes produced by the compiler and assembler are downloaded to a PROM-programmer for transfer to the 87C51FA on-board ROM.

The C-compiler is based on the traditional Kernighan and Richie (K&R) standard with additional implementation of the proposed ANSI-standard and Archimedes special purpose libraries. Source code is written with any standard ASCII editor and subsequently called by the compiler.

The Archimedes assembler accepts a source file written in Intel 8051 assembly language and generates an appropriate object file. As with C source code, any standard ASCII editor can be used to write the assembly program.

The Archimedes linker program is used to link object files from C and/or Assembly programs to one HEX load module suitable for transfer to an external PROM. Default libraries for high level languages and configuration setups are automatically loaded. Various output formats are available such as BINARY, ASCII, or HEX for Motorola, Intel, Hewlett-Packard, etc. controllers. List and map files can be generated as options.

Downloading of the linked HEX file is accomplished using a PROM programmer and appropriate software from Needham Electronics. The programmer uses an option slot in an IBM/AT personal computer. A cable from the option board to an external socket links the programmable 87C51FA microcontroller to the AT computer. Data are downloaded in INTEL-STANDARD HEX format.

Note! Neither the Support Technician nor the Field Operator will normally have a need to employ the software development kit. The kit is required only if changes are made in the data acquisition software. Details of the development kit are included here to furnish a complete understanding of the MARK-1 system.

DEVELOPMENT SOFTWARE

All of the development software is installed in a directory called ARCH. ARCH is reached from the root directory. Subdirectories for compile, assemble, link, download, etc. are installed within ARCH. Note that the line editor, EDLIN, can be reached from ARCH.

This eliminates the need to write a program from the DOS directory and then transfer to ARCH for compiling.

MARK-1 CODE GENERATION

This section describes the various steps required to write programs, compile, assemble, link, and download program files to an 87C51FA microcontroller for a MARK-1 system.

1. Writing Programs, Routines, Subroutines

The programs, both C and Assembly routines, written for the MARK-1 use the standard IBM EDLIN line editor.

The Archimedes version of the C language used to write the MARK-1 routines is based on the K&R Standard which incorporates the proposed ANSI Standard rules along with several in-line functions developed by Archimedes for the INTEL 8051 family of microcontrollers. Examples of the in-line functions are READ_BIT; WRITE_XDATA; etc.

Assembly language routines were also written using the EDLIN line editor. Assembly language instructions are those specified by the INTEL Corporation for the MCS-51 series of microcontrollers.

2. Library Functions

The Archimedes Software Development Kit contains most C-library functions that apply to the Intel 87C51FA microcontroller. The libraries for the 87C51FA functions are in a file named CL8051S.R03.

Libraries exist for character handling, non-logical jumps, input/output, general utilities, string handling, and mathematics. To use any library function, an appropriate header file must be declared before any reference to the functions are made. Header files are declared with: `#INCLUDE <FILE>` statements at the beginning of C-language programs. The specific library file (i.e.: CL8051S) is specified in a command file, IDAHO.XCL. (See Linker, subsection 6, for details on the command file).

Specifying a library file in a command file will not link all possible library functions in a program. Those functions designated as "library" modules are loaded only if they are referenced in another module. User-written modules are "program" modules and are always linked when the filename is processed by the linker.

All modules in the library file CL8051S are "library" modules with the exception of the file CSTARTUP. The CSTARTUP file is always loaded during a linking operation.

The program, IDAHO.RL3, for the MARK-1 system declares two header files; i.e., `<i051.h>` and `<math.h>`. The `<math.h>` file includes libraries for computing logarithms, exponents, etc. The `<i051.h>` file defines I/O ports and I/O pins for the 87C51FA microcontroller.

3. CSTARTUP

CSTARTUP is the first program module called by the linker during linking operations. It is an assembly language program, file extension of .S03, that contains the startup routine and defines a number of system functions. These include:

1. define memory model,
2. allocation of stack segments,

3. register bank definition,
4. initialization of stack and registers,
5. user-specified interrupt handlers,
6. call to main () program.

CSTARTUP must be written for a particular microcontroller and program. If a new CSTARTUP is written, or an existing CSTARTUP is modified, the new version must be assembled using the Archimedes assembler. After assembly, the CSTARTUP program is stored in the CL8051S library file. The procedure to store CSTARTUP is as follows:

Note! CPU prompts are underlined.

```
C:\ARCH>      XLIB
_*            DEF-CPU 8051
_*            REP-MOD CSTARTUP CL8051S
_*            EXIT
```

These steps will replace the existing CSTARTUP file in the CL8051S library with a new or modified version.

The utility program, XLIB, is designed to aid in the maintenance or creation of relocatable library modules. For further information on the CSTARTUP file, the reader is referred to the *Archimedes Development Kit*, Section T.8.1, "Modifying and Assembling CSTARTUP."

4. C-Compiler

The C-compiler is a program that accepts C language source files and attaches header files and generates a relocatable object file with extension type .R03, which can be processed by the linker. The output file is in an Archimedes proprietary format, UBROF (Universal

Binary Relocatable Object Format). When the compiler is called, a number of option switches are available to produce various listing files. The option switches are defined in the *Archimedes Development Kit* manual.

The call to the compiler for the MARK-1 system is the following:

Note! CPU prompts are underlined.

C:\ARCH> C-51 IDAHO.RL3 -e -ms -P -L -q -g

where C-51 is the execute file for the compiler,

IDAHO.RL3 is the C-language program to be compiled.

The remaining argument list are option switches:

- e enables Archimedes extensions to ANSI-C language
- ms specifies device memory size
- P generates PROMable code
- L generates LIST file
- q places mnemonics in LIST
- g enables global checks

5. Assembler

The Archimedes assembler is a two-pass assembler that operates as a single program; i.e., no temporary files are generated during execution. The assembler accepts source files written in Intel 8051 assembly language (file extension .S03). The generation of a list file is an option specified by the operator (file extension .LST). The object file generated has the same name as the source file with a default extension .R03. A number of assembler options are available. (See the *Archimedes Development Kit* manual for a list of options.)

The call to the assembler for the MARK-1 system is the following:

Note! CPU prompts are underlined.

C:\ARCH> a8051 (operator enters the assembler execute file)

SOURCE FILE[.MSA/.S03] = (enter filename with .S03 extension)

LIST FILE [.LST] = (enter filename with .LST extension)

OBJECT FILE [FILENAME.r03] = (computer displays same filename
as the source file with a default extension of .r03)

OPTIONS = S X

where S option puts all symbols in object code

and X option produces a cross reference and symbol table.

The assembler generates an object file in UBROF format, to be processed by the Archimedes Linker. The UBROF format is an Archimedes proprietary format, University Binary Relocatable Object Format.

6. Linker

The Archimedes linker is used to link together a number of object files generated from compiled and/or assembled programs to generate a load module suitable for downloading to a PROM. In addition to object files, default libraries are automatically loaded.

The standard command line for the linker starts with XLINK, which calls the linker.

XLINK [-OPTIONS] INPUT FILE(S) [-OPTIONS]

There is a limited amount of space available on the command line. Therefore, for a large number of options, input files, etc., an option exists to allow a call to a "command file", rather than using a command line. The option -f, calls a specified command file. The file has an extension of .XCL. To link the programs for the MARK-1 system, the command line used is:

XLINK -f IDAHO.XCL

where: **XLINK** is the call to the linker,
-f is the "call command file" option,
IDAHO.XCL is the command file name.

The command file lists all options and object files to be linked, including library files.

The output of the linker is a hex file with extension **.HEX**, that is in a suitable format for downloading to a PROM. Downloading to an INTEL 87C51FA uses the "INTEL-STANDARD" hex format. The format is defined in the *Archimedes Development Kit* manual.

7. Summary of Software Development

This section lists the computer operations required to compile, assemble, and link all of the routines required for the MARK-1 system.

The various programs required for the MARK-1 system include:

IDAHO.RL3	The primary C-language program.
EVENT1.S03	An interrupt routine written in assembly for data acquisition.
RDET1.RL2	A data acquisition subroutine written in assembly and called from EVENT1.S03 . This routine reads the radiation detectors.
SETGAIN.S03	A subroutine called from EVENT1.S03 to check and change the system gain. This routine is written in assembly language.
CSTARTUP.S03	The startup routine is written in assembly and is stored in the library file.
IDAHO.XCL	The command file for the linker program. This file uses special code. No assembly or compilation is required.

CL8051S The library file for the 87C51FA microcontroller. The file is called in the IDAHO.XCL routine.

The steps required to prepare each program are given below:

(1) CSTARTUP.S03

- A - Write program in assembly language
- B - Assemble (see Subsection 5 above)
- C - Load to library file (see Subsection 3)

(2) IDAHO.RL3

- A - Write program in C language
- B - Compile (see Subsection 4)

(3) EVENT1.S03; RDET1.RL2, SETGAIN.S03

(For each program)

- A - Write programs in assembly
- B - Assemble (see Subsection 5)

(4) IDAHO.XCL

- A - Write program using specified code
(This is the command file for the linker.)

(5) LINKER

- A - Follow instructions in Subsection 6

8. Download to Microcontroller ROM

The process of compiling, assembling, and linking results in the generation of a load module suitable for downloading to a microcontroller ROM. The ROM for the 87C51FA is built into the microcontroller chip. The code for downloading is a HEX file in INTEL-STANDARD format. The format is defined in the *Archimedes Development Kit* manual. The LINKER program places the HEX code in a file named IDAHO.HEX. This filename is

specified in the linker command file, IDAHO.XCL. The "-F" option in the command file specifies the output format, INTEL-STANDARD. The "-O" option specifies the filename, IDAHO.HEX.

The hardware and software required to download the output file are products of Needham Electronics. The programmer hardware (Model PB-10) uses an option slot in an IBM AT personal computer. A cable from the option board to a special purpose module and socket links the 87C51FA chips to the computer. Note that the special purpose module is designed specifically for the 87C51FA.

The software program to control the programmer is installed in the ARCH directory under the subdirectory named EMP.

The "manual" for the programmer and software is available within the EMP program. To enter the program, type EMP after the computer prompt:

C:\ARCH> EMP

From the main menu, enter function key, F10, for HELP. The EMP HELP MENU will be displayed. Various help instructions are available. If a full copy of the "manual" is required, select option 6, "EXTENDED HELP". From the EXTENDED HELP MENU, select option 6, "PRINTING EMP.HLP". Instructions for printing the manual are given.

The Model PB-10 hardware and EMP software constitute a general purpose PROM programmer designed to program a large number of PROMs available on the market from several manufacturers. The EMP software is menu driven and all necessary options are listed in the EMP MAIN MENU.

Because the only PROM of interest in this writeup is the on-board ROM of the INTEL 87C51FA, the steps here are those required to program the 87C51FA.

The simplest block diagram of the Model PB-10 programmer can be defined as a 3-block series diagram with the 87C51FA ROM on one end, the AT computer data file on the other end, and a buffer memory between. The buffer memory is physically located on the PB-10 option board.

Programming an 87C51FA is basically a two-step process of transferring a computer data file to the buffer memory, and then from the buffer memory to the 87C51FA ROM.

The following steps are required to program an 87C51FA ROM for use in the MARK-1 system. These steps assume that the program has been written, compiled and assembled, linked, and a HEX file is available for downloading. The HEX file normally used for the MARK-1 is IDAHO.HEX.

- (1) Call EMP program from the ARCH directory.
`C:\ARCH> EMP`
Computer screen displays main menu.
- (2) Select Manufacturer/Device (OPTION 5).
Computer displays selection of manufacturers.
Select INTEL (#14).
- (3) From Device Types:
Select 87C51FA (#50).
- (4) Computer displays main menu and STATUS/SETTINGS.
- (5) Verify that manufacturer, device and algorithm are correct.
i.e., INTEL; 87C51FA; QUICK PULSE.
- (6) Verify BUFFER and DEVICE are each "000000, 001FFF".

Select options K and L respectively to change BUFFER AND DEVICE.

- (7) Verify option O:
Splitting: 1 Device #: 1
- (8) Verify option p
of sets: 1 Set #: 1
- (9) Verify option Q:
Pgm Vpp: 12.50 V Pgm Vcc: 5.00 V
- (10) Verify option R:
Verify H: 5.00 V Verify L: 5.00 V
- (11) Verify option S:
Buffer: 000000, 001FFF
- (12) Verify option T:
File: 00000000, 00001FFF
- (13) Verify option U:
File Type: INTEL HEX
- (14) Verify option V:
Filename: IDAHO.HEX
- (15) Verify option W:
Port Base: 0300h
- (16) After verification of Status/Settings, computer file can be downloaded to buffer memory.
Select option 8 from main menu
(Load file from disk)
- (17) Buffer memory can be loaded to device
Select option 1 from main menu
(Program with selected algorithm.)

WARNING! FOLLOW DEVICE PLACEMENT EXACTLY.

- (18) After programming is complete:
Select option 3 from main menu
(Verify device equals buffer)

Device is now ready for installation in the MARK-1 system.

Note that details concerning the operation of the PB-10 Programmer and Software are available in the HELP (F10) menus.

5.1.2 Data Acquisition and Control Software

The primary program for the MARK-1 for data acquisition and control is written in the C language. Most subroutines are also written in C; however, the interrupt routine, EVENT1, the data read routine, RDET1.RL2, and automatic gain control routine, SETGAIN, are written in assembly language to increase the speed of these functions.

The microcontroller in the MARK-1 system is a type 87C51FA, a member of the Intel 8051 family of microcontrollers. The 87C51FA is characterized by an on-board 256-byte temporary storage data RAM, an 8-kbyte program ROM, 4 I/O ports and an RS232 serial port.

Additional support chips for the MARK-1 include analog multiplexers, a 12-bit analog-to-digital converter (ADC) for input of analog signals from eight radiation detector analog circuits and two temperature sensors, a real-time-clock system with date (month and day) and time (hours, minutes, seconds, and milliseconds), a digital-to-analog converter (DAC) for automatic gain control to the photomultiplier tube (PMT) high voltage bias supply, and logic

circuits for function decode and coincidence control. The logic circuits are implemented in field programmable gate arrays (FPGA) from XILINX Corporation. Data are stored in a 512-kbyte removable memory which is a credit card size plug-in CMOS memory device with a built-in battery. The card can be removed from the system and manually transported to another computer for data analysis. There is no on-board data analysis capability in the MARK-1 system.

This section briefly describes the on-board data acquisition software programs used in the MARK-1 system.

MARK-1 SOFTWARE OVERVIEW

The on-board software written for the MARK-1 system is designed for data acquisition and control. The software consists of a main program plus a number of subroutines. The subroutines are written to perform various initialization functions, select "TEST" or "RUN" mode, control data flow to or from the serial port and external memory, control the automatic gain function, and service interrupts.

The main program and most subroutines are written in C. The interrupt service routines are written in assembly language for the sake of speed. The main program and the C subroutines are written in a program file called IDAHO.RL3. The primary interrupt service routines are in a file called EVENT1. The assembly language subroutine named RDET1.RL2 is called from EVENT1 and is used to read the data from eight radiation detectors. The assembly language subroutine, SETGAIN, also called from EVENT1, is used for automatic gain control.

The C program file, IDAHO.RL3, follows a standard structure for C programs which consists of:

(1) Preprocessor Directives

The IDAHO.RL3 file contains two types of directives. The #DEFINE directive declares program constants. The constants defined in IDAHO.RL3 are TRUE = LOGIC 1 and FALSE = LOGIC 0. The #INCLUDE directives are used by the compiler to insert specified library functions in the program. The "header files" included in IDAHO.RL3 are I051.H, which defines the I/O ports of the system microcontroller, and MATH.H, which contains math routines such as sine, cosine, square root, etc.

(2) Global Variables

The variables are declared outside the bodies of all routines and as such are available to all routines. These variables may be single value variables such as MEMPOINTER and BANK, or arrays such as DET[16].

(3) MAIN() Function (see Flowchart Figure B.1)

The word MAIN in this context has a double meaning. The function is not only the primary (i.e., main) function, but by C language convention the main program is also named MAIN. This is the first function performed when the program is started. All other functions are called from MAIN.

The first portion of MAIN is a declaration of C subroutines used by the program. By declaring these subroutines in MAIN, the subroutines are available not only in MAIN, but also other subroutines that are called from MAIN. Note that these

subroutines are only declared within MAIN, but are not defined. Function definitions are given after the MAIN function.

(4) Other Functions

After the MAIN function, other functions are now defined. The definitions of functions consist of the series of C language instructions required to perform the given function. Including MAIN, there are a total of 17 C language routines (functions) defined for the IDAHO.RL3 program.

In addition to the C functions, there are three programs written in assembly language; i.e., EVENT1, RDET, SETGAIN. The first of the assembly language programs is the interrupt service routine, EVENT1 (see Flowchart Figure B.2). The only interrupt in the MARK-1 system is the receipt of valid coincidence pulses from the radiation detectors. The receipt of an interrupt transfers control to an "interrupt handler" (see the CSTARTUP program). The interrupt handler calls the program EVENT1. EVENT1 contains instructions to read all required data. These data include peak amplitudes from eight radiation detectors, date and time of the event, two temperature values, and automatic gain control information.

In EVENT1 a local call instruction, LCALL, to an assembly language subroutine, RDET, (see Flowchart Figure B.3) reads the outputs of eight radiation detectors. Subsequently, a LCALL to a C language subroutine, RDCLK, (see Flowchart Figure B.4) reads date and time from a real-time-clock chip (see the IDAHO.RL3 program). A LCALL to a C language subroutine, RDTEMP, (see Flowchart Figure B.4) reads temperature sensors. Finally, a LCALL to the assembly language subroutine, SETGAIN, (see Flowchart Figure B.5) performs tasks for automatic gain control and records data for gain control and subsequent data analysis.

Flowcharts for other C functions are also included in Appendix B. Listings for all these functions are given in the program listing for IDAHO.RL3.

In a manner similar to a C program, the assembly programs have a required structure. These structural requirements exist because of the use of the Archimedes Software Development System; specifically, the linker that ties all programs together. The basic structure is as follows:

(1) Directives

These are declarations that tell the linker what to do. Directives used in the MARK-1 assembly programs include:

PUBLIC (symbol): Makes these symbols (functions) available to other program modules.

EXTERN (symbol): Makes these external symbols defined in other modules available to this module.

RSEG CODE: Allows the linker to treat the following code as a "relocatable code segment". That is, the code does not have a predetermined location in memory.

(2) Assembly Language Program

This is the user-written assembly program.

Note: That in the program EVENT1, a directive, RSEG UDATA, appears prior to the END statement in the program. RSEG UDATA is a directive that defines the variables SCRATCH1 and SCRATCH2 as "relocatable data segments". Because these variables are defined within the assembly program, they are not available outside the program EVENT1.

FUNCTION DESCRIPTIONS

The MARK-1 software is written in four program files. The primary program file, IDAHO.RL3, is written in the C language and contains 17 program functions including MAIN. The second program file, EVENT1, is an assembly language program containing only one function: the interrupt service routine. The third and fourth program files are RDET1 and SETGAIN. Both are single function, assembly routines called from EVENT1. RDET1 reads eight radiation detector analog input signals, converts the signals to digital values, and temporarily stores the digital data in the on-board microcontroller random access memory (RAM). SETGAIN is an automatic gain function that checks for excessively high (saturated) or low (less than half maximum value) signals and adjusts system gain to optimize data values. Automatic gain information is stored in the data stream and subsequently used in external data analysis routines for data normalization.

For detailed information about the acquisition and control software the reader is referred to the flowcharts and program listings included in Appendix B.

5.2 Data Processing

5.2.1 Data Transfer

During data acquisition, the MARK-1 system stores data on a removable credit card memory. For analysis these data will need to be transferred from the removable memory to a computer memory. This transfer uses a commercial hardware and software package manufactured by Needham Electronics. The hardware consists of a Model SA-10 EPROM Programmer and a Model SAMC-10 plug-in adapter. The software package is installed in

the MARK-1 portable computer under the directory SA10PR. The Model SA-10 Programmer is designed to be used not only as a programmer, but also as a device to read data from a memory device (such as the removable credit card memory). It is this latter function which is used to transfer the data stored on the memory card to the computer for analysis.

The operation of the SA10PR software package is detailed in *Volume 2: Operations Manual for the MARK-1 Pulsed Ionizing Radiation Detection System*, Section 4.3.

The standard format for data stored on a memory card is a 4-bit hexadecimal character. In the MARK-1 application, there are 64 4-bit characters per data record and there is one data record for every radiation pulse recorded by the system. The MARK-1 memory card has a capacity of 512 kbytes of data. The MARK-1 data are stored on the memory card as 4-bit hexadecimal data for the date and time information and 8-bit binary data for the detector peak amplitudes, temperature and high voltage data. A total of 32 bytes of data are stored on the card for one record. This equates to a maximum of 16,384 records which can be stored on one card.

The transfer of data from the card to the computer is a serial transfer of ASCII characters. The hardware/software package reads the 32 8-bit bytes of data per record from the card, converts this data to 64 4-bit hexadecimal characters, converts the hexadecimal data to 64 8-bit ASCII characters, and serially transfers the ASCII data to the computer via an RS-232 serial port. Because of the conversion process, the 512 kbytes of data on the card results in 1024 kbytes of ASCII data stored in the computer.

5.2.2 Data Analysis Routines

The analysis software for the MARK-1 system has been written to perform the following analysis functions:

1. Process data to obtain peak amplitude values for each detector.
2. Calculate the energy and dose deposited in each detector.
3. Generate amplitude ratio graphs.
4. Process data for pulse height analysis (PHA).
5. Generate PHA or Amplitude versus Block No. graphs.

These functions include the following operator-selectable options:

1. Select data group (from a single data record to 1000 records) with which to obtain average amplitudes, etc.
2. Select either a linear or natural log display for the shielded detector-to-detector amplitude ratio graph.
3. Select whether ratio plots will or will not be included in the hard copy data.
4. Select either separate or overlay graphs for the unshielded and shielded detectors.

In the case where multiple records are selected for analysis (i.e., greater than 1 and less than or equal to 1000), the first step in the analysis is to normalize all the amplitude values to the lowest gain setting of the multiple record group. This is required because of the automatic gain capability of the MARK-1 which compensates for the widely varying intensities of the photon source and the fact that initially the gain of the system may not be

set so that all detectors are operating within their range. This can lead to different data records having different gain settings and therefore the data will need to be normalized for record to record comparisons or compilation. Once all records are normalized to the same gain setting, amplitude averages are calculated for each detector in the system. Calculations from that point on are performed using the detector average values.

If the detector-to-detector ratio data are to be printed with graphs, the data are fit to a linear function; that function is also plotted along with the ratio data on the respective graphs. The fit parameters are also printed with the data. Under the logarithm display option (shielded detector data only), the linear fit of the $\text{Ln}(\text{ratio})$ versus shield thickness is equivalent to an exponential fit.

If ratio data are printed without ratio graphs, the data for each record in the series is printed individually with no normalization performed and no average values taken.

There are two display options available in the PHA mode of analysis. The first option, called the Amplitude option, displays the peak amplitude data versus data record sequence number. For radiation sources that are repetitively fired at a constant rate, this type of plot shows the time variation of the source intensity. Under the Amplitude option, the detector number (or high voltage) must be selected. This option is especially valuable to either track the time variations of the intensity of the radiation source, or to use as a trouble shooting aid of the gain/high voltage electronics for the MARK-1 system.

The second option, called the PHA option, displays the familiar pulse height spectrum found in typical gamma-ray spectrometry. In this case the ordinate is the number of data pulses within each pulse height channel and the abscissa is the channel number. As with the Amplitude option, the PHA option requires that the detector number (or high voltage) must

be specified. An option in the PHA display allows for either raw or normalized detector data to be used.

For both the Amplitude and PHA display options, the operator has a choice of horizontal scaling factors. These include:

1. Full-scale display of 4096 channels.
2. Automatic horizontal scaling that displays multiples of 25 channels that include the lowest and highest channels with at least one count.
3. Expanded display that includes multiples of 25 channels with only the data from the channels between the lowest and highest channels requested being displayed. For example, if channels from 43 to 68 are requested for display in the expanded mode, the horizontal axis will show numbered channels from 25 to 75, but only data from channels 43 to 68 will be displayed.

The entire analysis software (main program plus subroutines) has been written in the QUICK BASIC language. In the paragraphs that follow a brief description of the main program and the subroutines are described. At the beginning of each description the flow chart figure number and the program line numbers are given. The reader may refer to the corresponding flow chart and to the program listing in Appendix B for more detailed information.

PROGSAM.BAS/ MAIN Program (Figure B.11, program lines 10-182, 610-800) The MAIN program performs fundamental initialization functions and setting function key definitions.

Function Key Definitions

F1 - Display Ratio Graph

F2 - Print Data

F3 - Compute PHA

F4 - Select PHA channel

F6 - Select System and Data File (Must be pressed first to start analysis.)

F10 - Exit

A WAIT loop is written into the MAIN program to allow operator selection of various routines. Upon completion of a requested analysis function, the program returns to the WAIT loop to allow the operator to select the same or a different analysis routine. A function key is designated by the MAIN program that allows the operator to change data files for analysis without restarting the MAIN program. All function key definitions are printed on the bottom of the computer screen. The analysis program is first started by pressing the function key, F6.

Select System and Filename Subroutine (Figure B.11, program lines 192-535) This subroutine prompts the operator to identify the system from which data were collected and the filename containing the data. Entry to this subroutine is via function key F6. These selections are necessary because the system gain settings are unique to the particular system from which the data were taken. Specifically, there are two models of detector system (i.e., MARK-1 and Prototype) and they use different gain settings, etc.

RATIO Subroutine (Figure B.11, program lines 850-1330) This subroutine is written to process the raw data from the eight detectors and produce average response data, equivalent dose data, detector-to-detector response ratio data; generate a graph of the unshielded detector response ratio data versus detector diameter; generate a graph of the shielded

detector response ratio data versus shield thickness; fit linear functions to each of the response ratio graphs and provide hard copies of the results. Entry to this subroutine is via function key F1.

The detector-to-detector ratios are calculated from the absolute amplitudes of detector pulse data. Two sets of detector-to-detector ratios are obtained. The first set is generated from the four unshielded detectors; each detector signal amplitude is compared to the amplitude from the largest diameter unshielded detector crystal (detector #4). The second set of ratios is obtained from absolute amplitude data from the shielded detectors and the detector amplitudes are compared with the amplitude of the detector with the thinnest lead shield (detector #5). Both sets of ratio data can be obtained from either a single record for a particular source pulse number within a specified file or from the average values from multiple records for a sequence of source pulses (up to 1000 pulses) within the file.

The equivalent energy and dose deposited in each detector is also calculated. As is the case for the amplitude ratio data, this is performed for either a single data record or for the averages of up to 1000 records.

Two linear, least square analyses are performed: the first is on the detector-to-detector ratio data versus the detector diameter for the unshielded detector response data. The second is on the detector-to-detector ratio data versus the shield thickness for the shielded detector response data. Slope and intercept values are calculated for each analysis and printed with the results. Two graphs are also presented in the hard copy results: the first displays the unshielded detector response ratio data versus the detector diameter and the corresponding linear fit to these data, and the second is the shielded detector response ratio data versus shield thickness and linear fit to these data.

Single Record Analysis - The **RATIO** subroutine will call the **COMPUTE COMBINED DATA** subroutine (line 3690). In this subroutine the operator will designate single record or multiple record analysis. If single record analysis is requested, the subroutine will return to the **Display Ratio Graph** subroutine and the operator will be asked to specify the sequence number of the data record to be analyzed. The **READ** subroutine (line 4420) is called which reads the ASCII data from the specified record and converts the data to decimal format. Control is then returned to the **RATIO** subroutine which calculates the full scale energy for the record using the gain step number specified for that record and equation 5.1 or 5.2.

For gain step number less than 2:

$$\text{Full Scale Energy (MeV)} = \frac{32,000}{2^{7-\text{Gain Step No}}} \quad (5.1)$$

For gain step number equal to or greater than 2:

$$\text{Full Scale Energy (GeV)} = \frac{32}{2^{7-\text{Gain Step No}}} \quad (5.2)$$

The energy deposited in each detector is calculated using the full-scale energy and the amplitude of the detector pulse using equation 5.3.

$$\text{Deposited Energy} = \frac{\text{Detector Peak Amplitude}}{4095} \text{ Full Scale Energy} \quad (5.3)$$

The equivalent dose for each detector is calculated from the deposited energy and the mass of the detector scintillator using either equation 5.4 or 5.5.

$$Dose = \frac{\text{Deposited Energy (MeV)}}{\text{Mass of Detector (grams)}} 1.60209 \times 10^{-8} \quad (5.4)$$

$$Dose = \frac{\text{Deposited Energy (GeV)}}{\text{Mass of Detector (grams)}} 1.60209 \times 10^{-5} \quad (5.5)$$

The operator is asked to specify separate or overlay ratio graphs. The PRINT RATIO DATA subroutine (line 1350) is called to print the tabular results to the computer screen and/or to the printer and the CALL RATIO PLOTS subroutine (line 1940) is called to draw the ratio graph(s). These two subroutines print the tabular data and the ratio graphs to the computer screen. The subroutine returns to the WAIT loop in the MAIN program at line 790.

Multiple Record Analysis - If, in the COMPUTE COMBINED DATA subroutine (line 3690), the operator selects multiple record analysis, the operator is then asked to select the number of records to be combined (up to 1000 records) and the sequence number for the first record of the data group. The READ subroutine is called which reads all of the requested records and returns to COMPUTE COMBINED DATA subroutine. The COMPUTE COMBINED DATA subroutine normalizes all data to the lowest gain step number found in the group of data records, calculates the normalized full-scale energy, the average pulse peak value for each detector, the average energy deposited in each detector, and the average dose as outlined in equations 5.1 - 5.5. A flag is set indicating that combined data are being used (i.e., COMBIN% = 1) and the process control is returned to the RATIO subroutine. The combined data are then printed to the computer screen or to the printer using PRINT RATIO DATA and CALL RATIO PLOTS subroutines. The subroutine then returns to the WAIT loop in the MAIN program at line 790.

PRINT RATIO DATA Subroutine (Figure B.12, program lines 1350-1380) This subroutine prints data to either the computer screen or the printer. If the ratio data graphs are to be presented, the data are printed to the computer screen and if no graphs are to be drawn, the data are printed to the printer. Data to be printed includes filename, record number(s), date, start and stop time, full-scale energy, raw and normalized pulse amplitude data, deposited energy per detector, dose per detector, pulse amplitude ratios, and standard deviations. For data output to the printer (no graph), this subroutine calculates full-scale energy, deposited energy per detector, dose equivalents, and ratios. All results with the exception of the detector-to-detector amplitude ratios are calculated in the previously called subroutines and the ratios are calculated in this subroutine. Entry to the subroutine is from either the RATIO subroutine or the PRINT RATIO DATA-NO GRAPH subroutine.

CALL RATIO PLOTS Subroutine (Figure B.12, program lines 1940-2030) This is a short subroutine whose primary function is to call additional subroutines to plot the ratio data. Subroutines PROCESS 'A' GRAPH and PROCESS 'B' GRAPH are called to process the ratio data for the unshielded detectors and shielded detectors, respectively. Lines 1980-2020 constitute a WAIT loop to allow the operator to decide what to do with the graphs displayed on the computer screen. For example, the operator may elect to output the graph to the printer (with PRTSC) or select any other available function key.

PROCESS 'A' GRAPH Subroutine (Figure B.12, program lines 2110-2589) This subroutine produces the ratio graph for the four unshielded detectors. The subroutine determines an automatic scaling factor for the vertical axis, defines the location for the graph on the computer screen, draws the graph axes and labels, plots data points, performs a linear least squares fit of the ratio data, calculates slope and intercept values, and draws the data plot line. Entry to this subroutine is via a call from the CALL RATIO PLOTS subroutine.

DRAW SLOPE LINE Subroutine (Figure B.13, program lines 2590-2780) This subroutine calculates a linear least squares fit to the ratio data for both the unshielded and shielded detectors. This includes the linear/logarithmic option for the shielded detectors. Based upon the intercept and slope values, the subroutine calculates two points on the graph perimeter and draws a straight line between the two points. The two points are each defined as x,y pairs. The subroutine is written to guarantee that the slope lines that are drawn do not extend beyond the boundaries of the graph boxes. The routine is divided into two main sections: the linear graph computation (for shielded or unshielded detectors), and the semilogarithm graph computation (for shielded detectors only). Entry to this subroutine is from either the PROCESS 'A' GRAPH or PROCESS 'B' GRAPH subroutines.

PROCESS 'B' GRAPH Subroutine (Figure B.14, program lines 2870-3236) This subroutine produces the ratio graph for the four shielded detectors. The subroutine determines an automatic scaling factor for the vertical axis, defines the location for the graph on the computer screen, draws the graph axes and labels, plots data points, performs a linear least squares fit of the ratio data, calculates slope and intercept values, and draws the data plot line. There are two options available in this subroutine; i.e., linear graph and semilogarithm graph. Entry to this subroutine is via a call from the CALL RATIO PLOTS subroutine.

PRINT DATA-NO GRAPH Subroutine (Figure B.14, program lines 3480-3660) This subroutine prints the detector response data, etc. on an individual shot basis. It can print the results of just one data record or a group of data records. The operator will specify the number of data records to be printed and the sequence number of the first data record in the group. It calls the READ, PRINT RATIO DATA, PRINT SLOPE/INTERCEPT-NO GRAPH subroutines to read the data records, generate the detector-to-detector ratio data, and fit the ratio data to a linear function, respectively. Entry to this subroutine is via the function key F2.

COMPUTE COMBINED DATA Subroutine (Figure B.15, program lines 3690-4210) This subroutine combines multi-record data into averages for each of the detector peak responses. The limit of the number of data records which can be handled is 1000 records. As explained in the Multi-record Analysis section above, this program normalizes the data in the group to the lowest gain setting for that group and then calculates the averages for the detector responses. Once the averages have been determined, the energy deposited in each detector, dose in each detector, and detector-to-detector response ratios are all based on average detector response data rather than the individual shot data. Entry to this subroutine is via a call from the RATIO subroutine.

READ Subroutine (Figure B.15, program lines 4420-4869) This subroutine reads a data record and then calls the Convert ASCII to Decimal subroutine to convert the data to decimal format. The input data for each detector are given in four bytes of ASCII characters. The READ subroutine converts the first two bytes of each detector datum to decimal, then converts the last two bytes, and then combines the resulting decimal values to obtain the final decimal number for that detector response. The same procedure is followed in converting the temperature data. Entry to the READ subroutine is via calls from RATIO, PRINT DATA-NO GRAPH, COMPUTE COMBINED DATA or COMPUTE PHA subroutines.

CONVERT ASCII TO DECIMAL Subroutine (Figure B.16, program lines 4900-4960) This subroutine converts the data which is ASCII format to decimal format. It first converts the ASCII characters to HEX characters, combines them, and then converts the result to decimal. This subroutine is called by the READ subroutine.

DRAW GRAPH Subroutine (Figures B.17A and B.17B, program lines 5170-5878) This subroutine locates the graph on the display, draws the graph axes, places grid marks along

the axes and labels the axes. Entry to this subroutine is via PROCESS 'A' GRAPH or PROCESS 'B' GRAPH subroutines.

FIT Subroutine (Figures B.18, program lines 5920-6060) This subroutine arrives at a least squares fit of a linear function to the data and returns to the calling routine the Y intercept and slope of the linear function. Entry to this subroutine is via calls from PROCESS 'A' GRAPH or PROCESS 'B' GRAPH subroutines.

PRINT SLOPE/INTERCEPT-NO GRAPH Subroutine (Figures B.18, program lines 7000-7270) This subroutine prints the Y intercept and slope for the least squares fit of the unshielded or shielded detector ratio data to a linear function when no graph is specified. It calls the FIT subroutine to obtain the intercept and slope for the unshielded detector ratio data and then repeats the process for the shielded detector ratio data. Entry to this subroutine is via a call from the PRINT DATA-NO GRAPH subroutine.

COMPUTE PHA Subroutine (Figure B.19, program lines 7300-7532) This subroutine gets the data for either pulse height analysis (PHA) or amplitude versus record number (referred to as block number in the program). It first prompts the operator to specify PHA or Amplitude versus Block Number analysis. It then asks the operator to specify the starting block number and the number of records to be included in the analysis. If PHA has been specified, the subroutine will ask the operator to specify whether the raw data or the normalized data should be used in the analysis. It utilizes the READ subroutine to get the data for either type of analysis. Entry to this subroutine is via the function key F3.

PRINT PHA Subroutine (Figures B.20A and B.20B, program lines 7540-9000) This subroutine produces the PHA graph or the Amplitude versus Block Number graph. The PHA graph is a histogram of the number of times a peak amplitude occurs within the data

group versus the peak amplitude value. The Amplitude versus Block Number is a line graph of the peak amplitude versus the sequence number of the data records in the group. Either type of graph is presented for a specified detector number or for the high voltage. This subroutine asks the operator to specify the detector number or the high voltage prior to beginning to plot the data. Entry to this subroutine is via the function key F4. Since the data for this subroutine is generated in the COMPUTE PHA subroutine, it must be called before this subroutine is called. However, the COMPUTE PHA subroutine processes the data for all eight detectors and the high voltage in a single operation; therefore, it need only be called once for a particular record group.

6. CALIBRATION OF THE MARK-1 SYSTEM

6.1 Introduction

This section contains the detailed calibration procedure for a MARK-1 system. Included is a brief description of that portion of a MARK-1 that is involved in the procedure. Details of system theory, circuit descriptions, and schematics can be found in other sections.

The MARK-1 is a gamma radiation detection system that uses eight BGO/PMT radiation detectors (see Section 3 - THEORY). There is designed into the MARK-1 an automatic gain feature that extends the usable dynamic energy range of the system from about three orders of magnitude to greater than five. This extension is achieved by having eight selectable gain settings or steps. Each gain step is selected for a maximum full-scale energy. The various full-scale energies extend from 250 MeV full-scale to 32 GeV full-scale. Each successive gain step doubles the full-scale energy of the preceding step (i.e.: 250, 500, 1000, 32,000 MeV). Full-scale energy is doubled for each gain step by cutting the system gain by 1/2 for each step.

A simplified block diagram of the portion of the MARK-1 detection unit involved in the calibration procedure is shown in Figure 6.1. The high voltage module supplies bias to eight detectors. Each detector feeds a charge-sensitive preamplifier, and subsequently a shaping amplifier, peak detector, analog multiplexer, buffer amplifier, analog-to-digital converter (ADC) and the system microcontroller. Eight peak detectors feed a common analog multiplexer. The microcontroller looks at the incoming data from the ADC, and based upon the amplitudes of the data, may make no change in system gain or lower or raise the gain. If the signal amplitudes require a gain change, the microcontroller generates an 8-bit digital word that is converted to an appropriate analog voltage through the digital-to-analog

Figure 6.1 Calibration Procedure Block Diagram.

converter and associated buffer amplifier. The analog voltage controls the output voltage from the high voltage module. System gain is controlled by the high voltage fed to the detectors.

6.2 Circuit Gains and Conversion Factors

Described below are the various circuit gains and conversion factors required for calculations involved in the calibration procedure.

1. The output signal from the shaping amplifier is linearly proportional to the energy deposited in the BGO detector crystal. For a given gain step, the maximum output voltage is proportional to the full-scale energy for that gain step. The maximum usable amplitude from the shaping amplifier is +8.46 V. This is set by the maximum linear output available from the peak detector, which has a gain of +1 V/V.
2. The ADC is a 12-bit ADC that converts an incoming analog voltage to a 12-bit digital word. The ADC then has as an output a digital word with $2^{12} = 4096$ possible values (0 to 4095). The maximum analog input voltage that can be linearly converted is +4.82 Vdc which is set by an external reference voltage. This maximum value must be below the voltage supply to the ADC of +5 Vdc.
3. The gain of the multiplexer/buffer amplifier is selected to convert the +8.46-V maximum output from the peak detector to the +4.82-V maximum input to the ADC. This gain is approximately $+0.56974 (= 4.82/8.46)$.

4. The 8-bit digital control word from the microcontroller to the DAC to control the high voltage module has a maximum value of $2^8 = 256$ possible values (0 to 255).

The gains given above result in the following conversion factors.

1. Shaping amplifier output, volts as a function of deposited energy (e.g.: energy peak of a selected isotope):

$$\text{Volts Out} = \frac{\text{Energy Deposited}}{\text{Full Scale Energy}} 8.46 \text{ Volts.}$$

2. ADC output in counts as a function of deposited energy:

$$\text{Counts} = \frac{\text{Energy Deposited}}{\text{Full Scale Energy}} 4095.$$

3. ADC output in counts as a function of shaping amplifier output:
in volts:

$$\text{Counts Out} = \frac{\text{Volts Out}}{8.46} 4095.$$

6.3 Calibration Procedure

CAUTION

ONLY QUALIFIED PERSONNEL SHOULD PERFORM THIS CALIBRATION

The calibration procedure outlined below is that total procedure required for either (1) initial calibration of a new system, or (2) complete calibration of an existing system if the high voltage module or a BGO/PMT detector assembly has been repaired, replaced, or modified, or (3) a normal or preventive maintenance type calibration is required.

6.3.1 Equipment Required

1. **MARK-1 SYSTEM**

2. **EXTERNAL HIGH VOLTAGE CONTROL BOX**

This is an external control circuit that replaces the DAC in the high voltage control circuit. The box contains a DAC and eight switches. The box allows the operator to manually input the digital control word from the switches. The in-circuit DAC is removed from the logic board and the external box is plugged into the DAC socket via a ribbon cable.

3. **OSCILLOSCOPE**

Calibration must be up-to-date.

4. **MULTIMETER**

Calibration must be up-to-date.

5. EXTERNAL BROAD-BAND AMPLIFIER
(ORTEC Model 450 research amplifier or equivalent) Capable of adjustable gains from 10 to 2000; bandwidth to 1 MHz; no shaping time constant; unipolar output.

6. MULTICHANNEL ANALYZER (MCA)
(This procedure was developed using a Compaq portable computer with an Accuspec Plug-in MCA card.)

7. COMPAQ LTE-286 COMPUTER
(Standard external system computer)

8. ASSORTED CABLES

9. CHECK SOURCES
Recommended sources are Cs-137 (gamma peak at 661.7 keV) and Th-232 (gamma peak at 2614.6 keV). Co-60 is not recommended because of the difficulty of resolving the two gamma peaks at 1173.2 keV and 1332.5 keV.

6.3.2 Procedure

NOTE: For a "normal" or "preventive maintenance" type calibration, not all steps given below are required. For these steps, a note specifying "NOT REQUIRED FOR NORMAL CALIBRATION" will precede the instructions. Such steps are required if a high voltage module is repaired, modified, or replaced, or if a detector assembly is replaced.

1. Verify that system power is "OFF". Use POWER ON TEST SWITCH/LED for verification.

2. NOTE: NOT REQUIRED FOR NORMAL CALIBRATION

Temporarily remove the high voltage module from the system, remove cover(s). Remove R3 (see schematic and board layout in Appendix A, Figure A.4) and replace R3 with a 50-k Ω pot. Set the pot to approximately midpoint. Reinstall the high voltage module in the system such that the 50-k Ω pot is accessible for adjustment.

CAUTION: The potential voltage in this module can reach 1500 Vdc.

Care must be taken to avoid shock. Care must be taken to prevent shorting the module boards to ground.

3. Temporarily remove the DAC (Type AD557) from the logic board and plug the external high voltage control box into the DAC socket. The box plugs in via a ribbon cable. Verify that pin #1 of the box cable plug connects to pin #1 of the DAC socket.
4. Set all high voltage module potentiometers (8 each) to approximately midpoint on the pots. These pots are mounted on the top of the module.
5. Set each of the eight potentiometers on the detector assemblies to midpoint. These pots are located on the top of the four shielded detector assemblies and just above each of four unshielded detector assemblies.

6. Set all switches on the external high voltage control box to the "ON" or LOGIC 1 position.
7. Verify that the BACKUP BATTERY switch located at the top of the logic board is ON. The ON position is to the right.
8. The following steps are to be performed for each of eight amplifier (front-end) modules.
 - a. Remove an amplifier module from the assembly and remove it's cover plate. Leave all cables connected. Verify that the amplifier board will not be shorted to ground.

NOTE: (Ce-137 MORE APPLICABLE SOURCE FOR THIS STEP.)

- b. Place a Cs-137 source near the detector associated with the amplifier board.
- c. Connect oscilloscope probe to associated SHAPING AMPLIFIER MONITOR JACK on control panel.
- d. Turn system power ON. Enter the control word from the high voltage control box by depressing the momentary contact push-button switch on the control box. Verify that pulses are being displayed on oscilloscope.
- e. Adjust the DC offset (R10) and pole/zero (R3) pots on the amplifier board for 0 V offset and optimum damping. See the schematic diagram and component layout diagram, Figures A.7 and A.8 in Appendix A.

- f. Connect scope probe to the high gain amplifier output (TP2).
 - g. Adjust the high gain amplifier DC offset pot (R38) for 0 V offset.
 - h. Connect scope probe to the trigger amplifier output (TP4).
 - i. Adjust the trigger amplifier DC offset pot (R23) for 0 V offset.
 - j. Monitor the PEAK DETECTOR chip (type PKD01) pin 10 with voltmeter.
 - k. Adjust the peak detector comparator level pot (R26) to obtain 80 ± 3 mV at pin 10.
 - l. Turn system power OFF.
 - m. Replace cover plate on amplifier module and reinstall in place.
 - n. Repeat steps a-m above for all eight amplifier modules.
9. Select a check source. The preferred source is Cs-137 because the energy peak is relatively easy to read and measure on an oscilloscope. The energy peak for Cs-137 is 661.7 keV. Calculate the amplitude of the pulse at a shaping amplifier output for the check source being used. The full-scale energy will be 250 MeV. If Cs-137 is used, the amplitude will be ~ 22.4 mV. If another isotope is used, the peak amplitude is calculated as:

$$\frac{\text{Peak Energy}}{\text{Full Scale Energy}} 8460 \text{ (milliVolts)}.$$

10. NOTE: NOT REQUIRED FOR NORMAL CALIBRATION

Monitor the outputs of each of the shaping amplifiers at the control panel with a calibrated oscilloscope. Record the amplitudes. As each of eight amplifiers is monitored, move the check source near the associated detector to get optimum results.

11. NOTE: NOT REQUIRED FOR NORMAL CALIBRATION

Select a shaping amplifier whose amplitude is near the middle of the range of recorded amplitudes. For example, if Cs-137 is used as a check source, and the recorded amplitudes range from 20 mV to 40 mV, select an amplifier whose amplitude is near 30 mV.

12. NOTE: NOT REQUIRED FOR NORMAL CALIBRATION

Monitor the output of the selected amplifier on the oscilloscope. Adjust the 50-k Ω pot previously installed in the high voltage module to obtain the calculated amplitude from Step 9 above. For example, the calculated amplitude for Cs-137 is ~ 22.4 mV.

13. Turn system power OFF.

14. NOTE: NOT REQUIRED FOR NORMAL CALIBRATION
Measure the resistance value at which the 50-k Ω pot has been set. Replace the pot with a 1% resistor that is as close as possible to the measured value. Reassemble the high voltage module and reinstall in the system.
15. Turn system power ON. Enter the control word from the high voltage control box by depressing the push-button switch on the box.
16. Monitor the output of the #3 shaping amplifier on an oscilloscope using the check source (Step #9 above). Adjust the high voltage potentiometer for #3 located on the top of the high voltage module to obtain the peak amplitude calculated in Step #9 above. This adjustment must be as close as possible to the calculated value.
The accuracy of the calibration depends on this adjustment.
17. Turn system power OFF.
18. The remainder of this procedure requires the use of the external broad-band amplifier, the multichannel analyzer (MCA) and the COMPAQ LTE-286 system computer. Connect the system as shown in Figure 6.2.
19. The test setup in Figure 6.2 is used for the remainder of the calibration procedure. This procedure assumes that the external amplifier meets the requirements in the equipment list and the MCA has been adjusted for zero energy in channel zero.
20. Use the procedure outlined in the *Volume 2: Operations Manual for the MARK-1 Pulsed Ionizing Radiation Detection System*, Section 4.2, to turn ON and INITIALIZE the MARK-1 system. Note that the external high voltage control

Figure 6.2 Calibration System Component Connections.

box is still plugged into the MARK-1 detection unit such that gain control is from the external control box. The purpose for the initialization is to record data available only from the computer during the calibration procedure. Unless problems occur, there should be no need to turn system power OFF during the balance of the calibration procedure.

21. Verify that the control word switches at the external high voltage control box are all set to LOGIC 1. Enter the control word with the push-button switch. Repeat Step #16 above and readjust the #3 potentiometer if required.
22. The following sources and MCA conversion gain are recommended:
 - a. At least two check sources should be used. Recommended sources are Cs-137 and Th-232.
 - b. An MCA conversion gain of 4 K channels to convert at least 8 V input.
23. Verify that the test setup in Step #18 is connected.
24. Calculate the approximate gain of the external amplifier required to achieve an MCA conversion of 1 keV/channel. The approximate gain can be calculated as:

$$Gain = \frac{ENERGY\ PEAK \times MCA\ MAX\ INPUT}{PEAKAMPLITUDE \times MCA\ CONVERSION\ GAIN} ,$$

where:

ENERGY PEAK = Energy in keV for the peak of any available isotope.

MCA MAX INPUT = Maximum input in mV that can be linearly converted by the MCA.

PEAK AMPLITUDE = Amplitude in mV of the isotope peak. See Step #8 for calculated amplitude.

MCA CONVERSION GAIN = Number of MCA channels. (4096 channels recommended.)

For example, assume the following:

- a. Cs-137 is used - energy peak is 661.7 keV and calculated amplitude = 22.4 mv.
- b. The MCA used has a maximum input of +8 V (8,000 mV) and a conversion gain of 4 K (4096) channels.

Then:

$$\text{Amp Gain} = \frac{661.7 \times 8000}{22.4 \times 4096} = 57.7 .$$

25. Adjust the external amplifier for the approximate gain calculated above.
26. Select a source with the highest available energy peak. The 2614.6 keV peak from Th-232 is satisfactory. Obtain a pulse height spectrum from shaping amplifier #3.
27. If the energy peak does not fall into the correct MCA channel, where the channel number = energy in keV, then readjust the amplifier gain until the correct MCA channel is obtained. (NOTE: MCA channel for the 2614.6 keV thorium peak is channel 2615.)
28. For each remaining shaping amplifier, perform the following:
 - a. Obtain pulse height spectrum using same isotope as that used above for amplifier #3.
 - b. For each amplifier, adjust the corresponding high voltage potentiometer located at the top of the high voltage module to place the energy peak in the appropriate MCA channel. (Same MCA channel as that for amplifier #3 in Step #27 above.)

NOTE: This completes the calibration procedure for a normal calibration. The following additional steps are required for an initial system calibration or those cases where either the high voltage module or a detector assembly is repaired, replaced, or modified.

29. Connect a voltmeter to the "HIGH VOLT" monitor test jack on the control panel. See the schematic drawing, Figure A.14 in Appendix A, for the exact location of the monitor test jack. The signal at this point is 1 mV per volt of high voltage.
30. Connect the shaping amplifier #3 output at the control panel to the external amplifier.
31. Turn on the #3 coincidence select switch on the control panel. Verify that data are being displayed on the LTE-286 computer screen. If data are not being displayed, repeat Step 20 (MARK-1 initialization). Data should appear as:

```
FF MM DD HH MM SS SS SO
XX XO XX XO XX XO XX XO
XX XO XX XO XX XO XX XO
YY YO YY YO ZZ ZO OO FF
```

where line 1 is date/time data, lines 2 and 3 are detector data, line 4 is:

```
YY YO (2 places) are temperature data
ZZ ZO are high voltage data
OO FF are gain data.
```

Turn off the coincidence select switch after verification.

32. Verify that the remote high voltage control box switches are all set at LOGIC 1. Enter the control word by depressing the push-button switch on the control box.
33. Using the Th-232 check source, obtain a pulse height spectrum on the MCA. Verify that the 2614.6 keV energy peak is in MCA channel 2615 ± 5 channels. If the energy peak is not in the correct channel, repeat Steps #27 through #33.

NOTE: The procedure up this point has calibrated the system for the highest of the eight gain step. The balance of this procedure is to calibrate the system for the seven remaining gain steps. The objective is to determine the high voltage control words input from the external high voltage control box required to set the system gain for each gain step. Gains steps are numbered from #0 to #7. The highest gain is step #0 where the full-scale energy is 250 MeV. The system has been adjusted such that the 8-bit control word for the high gain has a decimal value of 255 (FF hexadecimal). Each increasing gain step (0 to 7) doubles the deposited energy required for a full-scale signal. This amounts to dividing the gain in half for each step. For a properly adjusted MCA, dividing the gain in half will drop the 2614.6 keV energy peak for Th-232 from a given channel number to one-half that channel number; for example, channel 2615 to channel 1307. The same logic follows for each succeeding gain step. For each gain step: the gain step number, the MCA channel number for the thorium peak, the control word, the voltmeter reading for the high voltage monitor, and the high voltage reading from the computer display shall be recorded as a permanent calibration record. Note that all adjustments are made for detector #3.

34. Record the required data for gain step #0. (See sample data format at end of procedure.)

35. Adjust the control word downward at the external high voltage control box. Enter control word with push-button switch at box. Take series of pulse height spectra with each input until the energy peak is in the correct MCA channel. Record the required data. Perform this process for each gain step.

NOTE: Using Th-232 for a check source will lower the MCA channel for each successive gain step. At gain step 3 or 4, the peak will be low enough that it cannot be resolved on the MCA display. When this occurs, leave the control word at the last valid gain step and adjust the external amplifier gain to place the peak in a much higher MCA channel. The MCA channel number for the next gain step can then be calculated from this higher channel. (See sample recorded data format at end of procedure.)

At about gain step 6, the amplitude of the thorium peak may be small enough that it cannot be resolved from amplifier noise. If so, the high voltage required for gain steps 6 and 7 can be estimated from plots of (1) $\log_{10}$ of full-scale energy versus control word, and (2) high voltage versus control word.

The control words can be extrapolated from the log energy versus control word plot. The corresponding high voltage can then be obtained from the high voltage versus control word plot.

Shown on the following pages are recorded calibration data and resultant plots for the calibration of the MARK-1 systems #1 and #2. The control words obtained from the calibration procedure are subsequently inserted in the on-board system software. If this is required, the hardware/software packages for generating data acquisition software are required.

Table 6.1

CALIBRATION DATA FOR MARK-1, SYSTEM 1 6/10/92

<u>GAIN STEP</u>	<u>MCA CH. Th-232</u>	<u>CONTROL WORD (HEX/DECIMAL)</u>	<u>HIGH VOLTAGE (VOLTMETER)</u>	<u>HIGH VOLTAGE - COMPUTER - HEXADECIMAL</u>
0	2615	FF/255	622	211
1	1307	D4/212	568	1E3
2	654	AD/173	519	1B9
3	327	89/137	473	193
4	164	68/104	432	16F

(Increase external amplifier gain to place energy peak in MCA channel 2615 - leave control word = 68/104.)

4	2615	68/104	432	16F
5	1307	4B/75	395	150
6	654	(2E/46)	358	12F
7	327	(13/19)	325	115

NOTE: Control words shown in () for gain steps 6 and 7 are extrapolated from gain plots.

Table 6.2

CALIBRATION DATA FOR MARK-1, SYSTEM #2

6/9/92

<u>GAIN STEP</u>	<u>MCA CH. Th-232</u>	<u>CONTROL WORD (HEX/DECIMAL)</u>	<u>HIGH VOLTAGE (VOLTMETER)</u>	<u>HIGH VOLTAGE - COMPUTER - HEXADECIMAL</u>
0	2615	FF/255	674	1E8
1	1307	D3/212	522	1BC
2	654	AC/172	477	195
3	327	88/136	434	170
4	164	67/103	397	152

(Increase external amplifier gain to place energy peak in MCA channel 2615 - leave control word = 67/103.)

4	2615	67/103	397	152
5	1307	49/73	362	134
6	654	(2A/42)	326	115
7	327	(12/18)	299	0FE

NOTE: Control words shown in () for gain steps 6 and 7 are extrapolated from gain plots.

For a normal calibration, the determination of the control words during calibration serve as a check that the system is operating properly. The system design is such that the operating range of the high voltage supply can vary widely. For example, the operating range of system #1 ranges from 325 V (low gain) to 622 V (high gain). System #2 operates from 299 V to 574 V. This is a function of the gains of the individual PMT's used in a given system. The control words for any system, however, should always be:

Gain <u>Word</u>	Control <u>Word</u>
0	FF
1	D4 $\pm$ 1
2	AC $\pm$ 1
3	88 $\pm$ 1
4	67 $\pm$ 1
5	4A $\pm$ 1
6	2C $\pm$ 1
7	13 $\pm$ 2

This occurs because any set of detectors used in a MARK-1 system will follow a given gain versus high voltage curve. Different sets of detectors will operate at different ranges along the curve. Adjusting the high voltage module sets the operating range. Then, for any operating range, the control for that range will be the same as any other range.

APPENDICES

**Appendix A
Circuit Schematic Drawings**

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

Figure A.1 Low Voltage Power Supply Schematic.

Figure A.2 Component Layout for Low Voltage Supply - Board #1.

Figure A.3 Component Layout for Low Voltage Supply - Board #2.

Appendices

NOTE: RESISTORS R4-R23 AND R31-R50 ARE 1 MEGOHM

COMPONENT LAYOUT FOR HIGH VOLTAGE BOARD #1

Figure A.5 Component Layout for High Voltage Supply - Board #1.

COMPONENT LAYOUT FOR HIGH VOLTAGE BOARD #2

Figure A.6 Component Layout for High Voltage Supply - Board #2.

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

Figure A.7 Analog Front-End Schematic.

Figure A.8 Component Layout for Analog Front-End Board.

Appendices

134

Figure A.10 Logic Board - Page 2 of 4 Schematic.

Figure A.11 Logic Board - Page 3 of 4 Schematic.

Appendices

138

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

Figure A.14 Monitor/Control Panel Schematic.

Appendix B
Software Flowcharts and Listings

Appendix B.1

Acquisition and Control Software Flowcharts

Figure B.1 Flowchart for: Acquisition Program MAIN().

Figure B.2 Flowchart for: Acquisition Function EVENT1.

Figure B.3 Flowcharts for: DUMP(), STORE(), RDET.

Figure B.5 Flowchart for: SETGAIN.

Appendices

Figure B.9 Flowchart for: READ().

FLOWCHARTS FOR
FUNCTIONS
CHGHV()
RXCHAR()
TXCHAR()
TXCHK()

Figure B.10 Flowcharts for: RXCHAR(),CHGHV(), TXCHAR(), TXCHK().

**Appendix B.2
Analysis Software Flowcharts**

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

Figure B.11 Flowcharts for: Analysis MAIN, SEL SYS/FILE, DISP RATIO GRAPH.

Figure B.12 Flowcharts for: PRINT RATIO DATA, CALL RATIO PLOTS, PROCESS 'A' GRAPH.

PROGRAM: PROGSAM.BAS
ROUTINE: DRAW SLOPE LINE

Figure B.13 Flowchart for: DRAW SLOPE LINE.

Figure B.14 Flowcharts for: PROCESS 'B' GRAPH, PRINT RATIO DATA-NO GRAPH.

Figure B.15 Flowchart for: COMPUTE COMBINED DATA.

Figure B.16 Flowcharts for: READ, CONVERT ASCII TO DECIMAL.

Figure B.17A Flowchart for: DRAW GRAPH - Part 1 of 2.

PROGRAM: PROGSAM.BAS
ROUTINE: DRAW GRAPH - PART 2 OF 2

Figure B.17B Flowchart for: DRAW GRAPH - Part 2 of 2.

Figure B.18 Flowcharts for: FIT, PRINT SLOPE AND INTERCEPT-NO GRAPH.

PROGRAM: PROGSAM.BAS
ROUTINE: COMPUTE PHA

Figure B.19 Flowchart for: COMPUTE PHA.

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

Figure B.20A Flowchart for: PRINT PHA Part 1 of 2.

PROGRAM: PROGSAM.BAS
ROUTINE: PRINT PHA - PART 2 OF 2

Figure B.20B Flowchart for: PRINT PHA Part 2 of 2.

**Appendix B.3
Acquisition and Control Program Listings**

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```

/*****
/*
/*      Idaho.r13      Data Aquisition System Control System
/*      Last Update: Jun. 18, 1990
/*      This program also uses the assembly language interrupt service
/*      routine contained in the file "EVENT1" which calls the assembly
/*      language subroutine contined in the file "RDET1"
/*      MODIFIED 05/21/92 FOR FIXED CONTROL WORDS BY R.S.L. for
/*      MARK 1 SYSTEM 1.
/*
*****/

#define TRUE 1
#define FALSE 0
#include <io51.h>
#include <math.h>

/*****
/*
/*      Global Variables
/*
*****/

unsigned int mempointer;      /*contains the lower 16 bits of the
                             /*credit card memory address

unsigned char bank,          /*contains the upper 4 bits of the
                             /*credit card memory address

        charcount,          /*contains the number of characters
                             /*output to the terminal

        det[16],            /*array to hold the values read by the
                             /*detectors

        temp[6],            /*array to hold the values read by the
                             /*temperature sensors

        terminal,           /*boolean variable, TRUE if the terminal
                             /*is connected, FALSE if not

        test,               /*boolean variable, TRUE if in test mode
                             /*FALSE if not

        hold,               /*contains the channel number of
                             /*detector being tested in test mode

```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```

        gnstep,          /*location in array contwd[8]          */
        manual,          /*auto/manual gain control select          */
        val1,
        hvcont,          /*contents of holding latch              */
        hivolt;          /*control word sent to dac              */

main()
{
    /*******
    /*
    /*      C Subroutines
    /*      This program also uses the assembly language interrupt service
    /*      routine "EVENT" which calls the assembly language subroutine
    /*      "RDET"
    /*
    /*******

    void init();          /*initializes ports, credit card memory
                        /*pointers, and interrupt control
                        /*registers

    void port();          /*initializes serial port to 19.2 kbaud,
                        /*1 start bit, 8 data bits, no parity,
                        /*and 1 stop bit

    void txchar(unsigned char data); /*transmits 1 character to the serial
                        /*port exactly as passed

    unsigned char rxchar(); /*returns 1 character from the serial
                        /*port exactly as received

    void rdclk();         /*reads the real-time clock and outputs */
                        /*to the subroutine "store" FF, month,
                        /*day, hour, min, sec, and ms

    void setclk();        /*sets real-time clock

```

```

void store(unsigned char data);    /*stores input byte to the credit card */
                                  /*memory and if "terminal" is TRUE, */
                                  /*sends the byte out the serial port, */
                                  /*formatted as a character. Formats */
                                  /*bytes into lines of 8 bytes, blocks of*/
                                  /*4 lines, and inserts all necessary CRs*/
                                  /*and LFs */
                                  */

void cal();                        /*puts system into test mode, sets */
                                  /*"test" to TRUE, loads "hold" with the */
                                  /*desired test channel, and enables the */
                                  /*event interrupt */
                                  */

void read();                       /*waits in loop until key is pressed, */
                                  /*then takes block address from serial */
                                  /*port, up to 5 digits, converts from */
                                  /*decimal, and outputs that block number*/
                                  /*from credit card memory */
                                  */

void stgn();                       /*store gain step to credit card */
                                  */

void selgn();                      /*select auto/manual and gain step */
                                  */

void chghv();

void txchk(unsigned char tsttdt);

/*****
/*
/*      main
/*
*****/

unsigned int i;
clear_bit(EA_bit);    /*disable interrupts */
test=FALSE;           /*set run mode */
init();
port();
output(P1,0xd0);
gnstep=read_XDATA(0);
output(P1,0x80);
manual=gnstep;
gnstep&=0x07;
manual>=4;
txchar(gnstep+'0');
txchar(manual+'0');

```

```

chghv();
output(P1,0x80);
clear_bit(RI_bit);          /*check to see if terminal connected      */
txchar('*');
for (i=0;i <= 20000;i++)
{if(read_bit(RI_bit))
{if(input(SBUF)==0x0d)
{clear_bit(RI_bit);
terminal=TRUE;
setclk();
cal();
selgn();
goto out;}
else terminal=FALSE;}
else terminal=FALSE;};
out:txchar('S');
set_bit(EA_bit);           /*enable interrupts          */
txchar('E');
check:while (!read_bit(RI_bit)); /*wait for keyboard input    */
clear_bit(EA_bit);         /*disable interrupts         */
clear_bit(RI_bit);
read();
set_bit(EA_bit);           /*enable interrupts          */
goto check;}

/*****
/*
/*      Function Definitions
/*
*****/

void init()
{output(P1,0x80);
output(P3,0xdf);           /*clear A20                  */
output(P0,0x00);
output(P2,0x00);
mempointer=0;              /*initialize credit card memory address */
bank=0;                    /*pointers                   */
clear_bit(IT0_bit);        /*make int0 level triggered   */
set_bit(EX0_bit);          /*enable int0                 */
clear_bit(EX1_bit);        /*disable int1                 */
set_bit(PX0_bit);          /*set int0 priority to high   */
output(P1,0x80);           /*start in initial state     */

void port()
{charcount=0;              /*initialize character count to zero */

```

```

output(PCON,0x80);          /*set double baud rate bit          */
clear_bit(EX1_bit);         /*disable external int1              */
clear_bit(ET1_bit);         /*disable timer 1 overflow interrupt */
output(TMOD,0x20);          /*set timer 1 as baud rate generator */
output(TH1,253);            /*load counter 1 for 19.2 kbaud      */
set_bit(TR1_bit);           /*turn on timer 1                    */
output(SCON,0x50);          /*set to mode 1, enable reception    */

void txchar(unsigned char data)
{clear_bit(TI_bit);
output(SBUF,data);
while (!read_bit(TI_bit));} /*wait for character to clear buffer */

unsigned char rxchar()
{while (!read_bit(RI_bit)); /*wait for incoming character */
clear_bit(RI_bit);
return(input(SBUF));}

void dump(unsigned char data)
{register unsigned char hold1,hold2;
hold1 = data > 4;           /*convert binary to character */
hold1&=0x0f;
hold1|=0x30;
if (hold1 > 0x39) hold1 += 0x07;
hold2 = data < 0x0f;
hold2|=0x30;
if (hold2 > 0x39) hold2 += 0x07;
txchar(hold1);
txchar(hold2);
txchar(0x20);               /*send a blank */
while (!read_bit(TI_bit));
if (charcount >= 7)         /*if charcount >= 7, send a CR LF */
{txchar(0x0d);
txchar(0x0a);
charcount=0;}
else charcount++;}

void store(unsigned char data)
{output(P1,((bank&0x0f)+0x90));
write_XDATA(mempointer,data);
output(P1,0x80);            /*set P1 to initial state */
if (mempointer==0xffff) bank++;
mempointer++;
if (terminal) dump(data);}

void rdclk()

```

```

{register signed char i;
unsigned char hold;
store (0xf1);
for (i=7;i>=0;i--)
    {if (i==5) i--;
    output(P1,(0xa0+i));          /*send address to real-time clock      */
    clear_bit(P3_7_bit);
    set_bit(P3_7_bit);
    output(P1,0xb0);              /*get data from buffer                */
    hold=read_XDATA(0);
    output(P1,0x80);              /*turn off buffer                    */
    set_bit(P3_5_bit);            /*set A20                            */
    output(P1,0xa4);              /*send status address                 */
    clear_bit(P3_7_bit);
    set_bit(P3_7_bit);
    clear_bit(P3_5_bit);          /*clear A20                          */
    output(P1,0xb0);              /*read buffer                        */
    if(read_XDATA(0))             /*if status bit set, do it again      */
        {output(P1,0x80);         /*set P1 to initial state            */
        /*txchar('*'); */
        i++;}
    else
        {output(P1,0x80);         /*if status bit clear, output the result */
        store(hold);};};}

void setclk()
{unsigned char j,hold1,hold2,hold3;
signed char i;
const unsigned char message[5][10]= {'E','n','t','e','r',' ','M','I','N',' ','',
'E','n','t','e','r',' ','H','R','S',' ','',
'E','n','t','e','r',' ','D','O','W',' ','',
'E','n','t','e','r',' ','D','O','M',' ','',
'E','n','t','e','r',' ','M','O','N',' ',''};

for (i=4;i>=0;i--)
    {if (i==2) i--;
    for (j=0;j<=9;j++)
        txchar(message[i][j]);    /*send out prompt                    */
    retry:output(P1,(0xa0+i+3));    /*send out address of current value  */
    clear_bit(P3_7_bit);
    set_bit(P3_7_bit);
    output(P1,0xb0);              /*read buffer                        */
    hold2=read_XDATA(0);
    output(P1,0x80);              /*set P1 to initial state            */
    set_bit(P3_5_bit);            /*set A20                            */
    output(P1,0xa4);              /*output address for status          */
    clear_bit(P3_7_bit);

```

```

set_bit(P3_7_bit);
clear_bit(P3_5_bit);          /*clear A20          */
output(P1,0xb0);              /*read buffer        */
if(read_XDATA(0))             /*if status bit is set, do it again */
{output(P1,0x80);              /*set P1 to initial state */
 /*txchar('*');*/
 goto retry;}
else
 {output(P1,0x80);              /*if status bit is clear, continue */
  hold1=hold2;                  /*convert current contents of clock */
  hold2&=0x0f;                  /*to ascii and display */
  hold2|=0x30;
  hold1>>=4;
  hold1&=0x0f;
  hold1|=0x30;
  txchar(hold1);
  txchar(hold2);};
while ((hold3=rxchar())!=0x0d) /*get new input, if any */
{txchar(hold3);                /*echo input to terminal */
 hold1=hold2;
 hold2=hold3;};
txchar(0x0d);                  /*send CR LF */
txchar(0x0a);
hold1<<=4;                      /*convert to BCD and send back to clock */
hold2&=0x0f;
hold3=hold1|hold2;
output(P1,0xb0);                /*write to buffer */
write_XDATA(0,hold3);
output(P1,(0xa0+i+3));          /*send address to clock */
clear_bit(P3_6_bit);            /*write to clock */
set_bit(P3_6_bit);
output(P1,0x80);};              /*set P1 to initial state */
output(P1,0xb0);                /*write 0 to buffer */
write_XDATA(0,0);
output(P1,0xa2);                /*zero seconds register */
clear_bit(P3_6_bit);
set_bit(P3_6_bit);
set_bit(P3_5_bit);              /*set A20 */
output(P1,0xa5);                /*send GO command */
clear_bit(P3_6_bit);
set_bit(P3_6_bit);
clear_bit(P3_5_bit);            /*clear A20 */
output(P1,0x80);};              /*set P1 to initial state */

void cal()
{signed char i,j;

```



```

register unsigned char temp;
const unsigned char message[4][15]= {'T','e','s','t',' ','M','o','d','e','?','(','Y','/', 'N',')'}
                                     'E','n','t','e','r',' ','C','h','a','n','n','e','l',' ','',
                                     'P','r','e','s','s',' ','a','n','y',' ','k','e','y',' ','t',
                                     'o',' ','s','t','o','p',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}

back:for (i=0;i <= 14;i++) txchar(message[0][i]); /*send prompt */
temp=rxchar(); /*get response */
txchar(temp); /*echo response */
txchar(0x0d); /*send CR LF */
txchar(0x0a);
if((temp=='N')||(temp=='n')) return; /*if "no" return to main program */
for (i=0;i <= 14;i++) txchar(message[1][i]); /*send next prompt */
temp=rxchar(); /*get response */
txchar(temp); /*echo response */
txchar(0x0d); /*send CR LF */
txchar(0x0a);
for (j=2;j <= 3;j++) /*send message */
for (i=0;i <= 14;i++) txchar(message[j][i]);
txchar(0x0d); /*send CR LF */
txchar(0x0a);
temp&=0x07; /*do modulo 7 division on selected channel */
output(P1,0x80); /*set P1 to initial state */
hold=temp; /*move selected channel to "hold" */
test=TRUE; /*set test mode */
set_bit(EA_bit); /*enable interrupts */
while(!read_bit(RI_bit)); /*wait for keyboard input */
clear_bit(EA_bit); /*disable interrupts */
clear_bit(RI_bit); /*clear data received bit */
test=FALSE; /*set mode to run */
output(P1,0x80); /*set P1 to initial state */
goto back;}

void rdtemp()
{register unsigned char i;
output(P1,0x08);
clear_bit(P3_6_bit);
set_bit(P3_6_bit);
for (i=0;i <= 2;i++)
{clear_bit(P3_3_bit);
set_bit(P3_3_bit);
output(P1,(((i+1) < 4)+0x08)&0x7f);
clear_bit(P3_6_bit);
set_bit(P3_6_bit);
while (read_bit(P3_4_bit));
temp[i*2]=read_XDATA(0);
temp[i*2+1]=read_XDATA(0);};

```

```
output(P1,0x80);}

void read()
{const unsigned char prompt[6]={'B','l','o','c','k',' ':' '};
signed char i;
unsigned char save,hold[6]={0,0,0,0,0,0};
unsigned int temp=0,power=1;
for (i=0;i<=5;i++) txchar(prompt[i]); /*send prompt */
while ((hold[5]=rxchar())!=0x0d) /*get response */
{txchar(hold[5]); /*echo response */
for (i=0;i<=4;i++) hold[i]=hold[i+1];}
txchar(0x0d); /*send CR LF LF */
txchar(0x0a);
txchar(0x0a);
for (i=4;i>=0;i--) /*convert decimal input to binary */
{temp+=(int)(hold[i]&0x0f)*power; /*block address */
power*=10;};
for(i=0;i<=31;i++) /*get bytes within the block */
{output(P1,(((temp>>11)&0x0f)+0x90));
save=read_XDATA((temp<<5)+i);
output(P1,0x80); /*set P1 to initial state */
dump(save);};}

void stgn()
{unsigned char numb;
numb=gnstep;
store(numb);
numb = hivolt;
store(numb);}

void chghv()
{
txchk(gnstep+'0');
if (gnstep==0x00) hivolt=0xff;
else if (gnstep==0x01) hivolt=0xd4;
else if (gnstep==0x02) hivolt=0xad;
else if (gnstep==0x03) hivolt=0x89;
else if (gnstep==0x04) hivolt=0x68;
else if (gnstep==0x05) hivolt=0x4b;
else if (gnstep==0x06) hivolt=0x2e;
else hivolt=0x13;
output(P1,0xe0);
write_XDATA(0,hivolt);
output(P1,0x80);
val1=manual<<4;
hvcont=val1|gnstep;
```

```

txchar(hvcont + '0');
output(P1,0xd0);
write_XDATA(0,hvcont);
output(P1,0x80);}

void selgn()
{unsigned char i,j,k,hold3;
const unsigned char msg1[2][37]=
    {'S','e','l','e','c','t',' ','a','u','t','o','/',
     'm','a','n','u','a','l',' ','g','a','i','n',' ','
     'c','o','n','t','r','o','l','(','a','/', 'm',') ',
     '.','S','e','l','e','c','t',' ','g','a','i','n',
     ' ','s','t','e','p','(','0','-','7',')','.',' ',
     '0',' ','=',' ','H','I','G','H',' ','g','a','i',
     'n','.'};
const unsigned char msg2[10]={'T','r','y',' ','a','g','a','i','n','!'};
for(i=0;i<=1;i++)
    retry: {for(j=0;j<=36;j++)
        txchar(msg1[i][j]);
        txchar(0xd0);
        txchar(0x0a);
        hold3=rxchar();
        txchar(0xd0);
        txchar(0x0a);
        if((hold3==0x41) || (hold3==0x61))
            {manual=0;
             goto kpgng;}
        if((hold3==0x4d) || (hold3==0x6d))
            {manual=1;
             goto kpgng;}
        if((hold3>=0x30)&&(hold3<=0x37))
            {gnstep=hold3-'0';
             txchk(gnstep+'0');
             goto kpgng;}
        for(k=0;k<=9;k++)
            {txchar(msg2[k]);
             txchar(0xd0);
             txchar(0x0a);
             goto retry;}}
kpgng:txchar(0xd0);
txchar(0x0a);}
chghv();
output(P1,0x80);}

void txchk(unsigned char tstdt)
{txchar(tstdt);}

```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
*****  
;  
;  
; EVENT1  
; Last Update: Jun 18, 1990  
; Interrupt Service Routine  
;  
; Modified 06/25/91 for PB-10 Programmer  
;  
*****
```

PUBLIC EVENT1
PUBLIC EVENT2

```
*****  
;  
;  
; External Variables  
;  
;  
*****
```

EXTERN test
EXTERN det
EXTERN temp
EXTERN hold
EXTERN bank

```
*****  
;  
;  
; External C Routines  
;  
;  
*****
```

EXTERN txchar
EXTERN store
EXTERN rdclk
EXTERN stgn
EXTERN rdtemp

```
*****  
;  
;  
; External Assembly Language Routines  
;  
;  
*****
```

EXTERN RDET
EXTERN SETGAIN

RSEG CODE

```

EVENT1:  clr IE.7;           Disable interrupts
          mov a,test;        Check to see if in run or test mode
          cjne a,#0,CALIB;    Jump to "CALIB" if in test
;
;      RUN MODE
;
          lcall RDET;         Read detectors
          mov r1,#0;
          lcall rdclk;        Read clock
          clr a;
          mov SCRATCH1,a;     ram ...
BACK2:    mov a,#det;
          add a,SCRATCH1;
          mov r1,a;
          mov a,@r1;
          mov r3,a;
          mov r1,#-1;
          lcall store;        ...to credit card memory
          inc SCRATCH1;
          mov a,SCRATCH1;
          cjne a,#10h,BACK2;
          mov r1,#0;
          lcall rdtemp;
          clr a;              Move data from temp sensors from internal
          mov SCRATCH1,a;     ram ...
BACK3:    mov a,#temp;
          add a,SCRATCH1;
          mov r1,a;
          mov a,@r1;
          mov r3,a;
          mov r1,#-1;
          lcall store;        ...to credit card memory
          inc SCRATCH1;
          mov a,SCRATCH1;
          cjne a,#06h,BACK3;
          mov r1,#0;
          lcall stgn;
          mov r1,#0;
          lcall SETGAIN;
          mov a,#bank;
          mov r1,a;
          mov a,@r1;
          cjne a,#08h,DONE2;
          ljmp DONE1;

```

```

DONE2:  ljmp DONE
;
;      TEST MODE
;
CALIB:  mov a,hold;          Get selected test channel
        swap a;
        anl a,#70h;
        mov P1,a;
        clr P3.6;          Latch into Mux
        setb P3.6;
        mov dptr,#0;
        clr P3.3;          Latch into A/D
        setb P3.3;
BACK1:  jb P3.4,BACK1;      Wait for A/D
        movx a,@dptr;
        mov SCRATCH1,a;
        movx a,@dptr;
        mov SCRATCH2,a;
        mov a,SCRATCH1;    Convert first nibble to ascii...
        swap a;
        anl a,#0fh;
        orl a,#30h;
        mov r3,a;
        clr c;
        subb a,#3ah;
        jc ARND2;
        mov a,r3;
        add a,#07h;
        mov r3,a;
ARND2:  mov r1,#-1;
        lcall txchar;      ...and send to terminal
        mov a,SCRATCH1;    Convert second nibble to ascii...
        anl a,#0fh;
        orl a,#30h;
        mov r3,a;
        clr c;
        subb a,#3ah;
        jc ARND3;
        mov a,r3;
        add a,#07h;
        mov r3,a;
ARND3:  mov r1,#-1;
        lcall txchar;      ...and send to terminal
        mov a,SCRATCH2;    Convert third nibble to ascii...
        swap a;
        orl a,#30h;

```

```
        mov r3,a;
        clr c;
        subb a,#3ah;
        jc ARND5;
        mov a,r3;
        add a,#07h;
        mov r3,a;
ARND5:  mov r1,#-1;
        lcall txchar;          ...and send to terminal
        mov r3,#0dh;
        mov r1,#-1;
        lcall txchar;          Send CR
        mov r3,#0ah;
        mov r1,#-1;
        lcall txchar;          Send LF

DONE:   setb IE.7;
DONE1  ret;

EVENT2: ret;                  Dummy service routine to retain compatibility

        RSEG UDATA
SCRATCH1: DS 1
SCRATCH2: DS 1

END
```

```

;*****
;
;      RDET1.rl2
;      Last Update: Jun. 18, 1990
;      Detector Reading Routine
;
;      Modified 06/25/91 for PB-10 Programmer
;
;*****

```

PUBLIC RDET

```

;*****
;
;      External Variables
;
;*****

```

EXTERN det

RSEG CODE

```

RDET:  mov P1,#0;          Latch first channel into Mux
      clr P3.6;
      setb P3.6;
      clr a;
      nop;
      nop;
      nop;
      mov r4,a;
BACK1: clr P3.3;          Latch into A/D
      setb P3.3;
      mov a,r4;          Calculate next channel...
      inc a;
      swap a;
      anl a,#70h;
      mov P1,a;
      clr P3.6;          ...and latch into Mux
      setb P3.6;

```



```
WAIT:    jb P3.4, WAIT;      Wait for A/D
          movx a, @dptr;     Read A/D
          mov r2, a;
          mov a, r4;
          rl a;
          add a, #det;       Move data into internal ram
          mov r1, a;
          mov a, r2;
          mov @r1, a;
          movx a, @dptr;
          mov r2, a;
          inc r1;
          mov a, r2;
          mov @r1, a;
          inc r4;
          cjne r4, #08h, BACK1;
;
          setb c;
          jc ARND;
ARND:    clr c;
          mov P1, #080h;
          ret;

END
```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
*****  
;  
;  
;      SETGAIN.S03  
;      18 JUNE 1990  
;      MODIFIED 06/26/91 FOR PB-10 PROGRAMMER  
;  
;  
*****
```

PUBLIC SETGAIN

```
*****  
;  
;      EXTERNAL VARIABLES  
;  
*****
```

```
EXTERN manual  
EXTERN det  
EXTERN gnstep
```

```
*****  
;  
;      EXTERNAL ROUTINES  
;  
*****
```

```
EXTERN chghv  
EXTERN txchk
```

RSEG CODE

```
SETGAIN:  mov a,manual;  
          add a,#35h;  
          mov r3,a;  
          mov r1,#-1;  
          lcall txchk;  
          mov a,manual;  
          cjne a,#00h,OKAY;  
          clr a;  
          mov r4,a;  
          mov a,#det;  
          mov r1,a;
```

```
CHECK:  mov a,@r1;
        inc r1;
        cjne a,#0ffh,NXT;
        mov a,@r1;
        cjne a,#0f0h,NXT;
        mov a,gnstep;
        inc a;
        mov gnstep,a;
        anl a,#08h;
        cjne a,#08h,CHGGN;
        mov a,#07h;
        mov gnstep,a;
        ljmp CHGGN;
NXT:    inc r4;
        inc r1;
        cjne r4,#04h,CHECK;
        mov a,r1;
        subb a,#02h;
        mov r1,a;
        mov a,@r1;
        anl a,#080h;
        cjne a,#00h,OKAY;
        mov a,gnstep;
        anl a,#0ffh;
        cjne a,#00h,INCGN;
        ljmp OKAY;
INCGN:  dec gnstep;
CHGGN:  mov r1,#0;
        lcall chghv;
OKAY:   ret;
END
```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
;-----;
;
;      CSTARTUP.S03
;
; This module contains the 8051 C startup-routine and
; must usually be tailored to suit customer hardware.
; Version: 2.00 [A.R. 10/Jun/87]
;-----;
      NAME      CSTARTUP

$DEFMM.INC                      ; define memory model

      PUBLIC   exit
      EXTERN   _R              ; Register bank (0, 8, 16 or 24)
      EXTERN   main
      EXTERN   EVENT1

;-----;
; Virtual stack segment. It should be mapped into external RAM
;-----;
      IF          single_chip = 0

      RSEG       XSTACK
      DS         512          ; Change if needed
xstack_end:
      ENDIF

;-----;
; Internal stack used for LCALL's and temporary storage for code generator help-routines
; (math etc). Stack can be located anywhere in the internal RAM with the exception of
; _R - _R+7 that are reserved for R0-R7. Note that C interrupt routines require that you
; increase stack by 25 bytes to assure that no overflows occur.
;-----;
      RSEG       ISTACK
stack_begin:
      IF          single_chip
      DS         150
      ELSE
      DS         45
      ENDIF
```

```

RSEG    CSTART
startup:                ; Should be at location zero
    SJMP    init_C
    ORG     startup+03H
    SJMP    handlr1
    ORG     startup+30H

```

```

handlr1:
    PUSH    ACC
    PUSH    B
    PUSH    PSW
    PUSH    DPH
    PUSH    DPL
    PUSH    _R+0
    PUSH    _R+1
    PUSH    _R+2
    PUSH    _R+3
    PUSH    _R+4
    PUSH    _R+5
    PUSH    _R+6
    PUSH    _R+7
    MOV     R1,#0
    LCALL   EVENT1
    POP     _R+7
    POP     _R+6
    POP     _R+5
    POP     _R+4
    POP     _R+3
    POP     _R+2
    POP     _R+1
    POP     _R+0
    POP     DPL
    POP     DPH
    POP     PSW
    POP     B
    POP     ACC
    RETI

```

```

;-----;
; This is the place to insert interrupt vectors/handlers in.      ;
; For locations consult the Intel Microcontroller Handbook.      ;
;-----;

```

```

init_C:
;-----;
; Ativate the (at link-time) selected register bank.
;-----;

    MOV     A,#_R
    MOV     C,ACC.3
    MOV     PSW.3,C
    MOV     C,ACC.4
    MOV     PSW.4,C

    MOV     SP,#stack_begin ; From low to high addresses
;-----;
; If you don't want global/static C variables to be initialized at startup
; you can just remove the next two lines.
; NOTE: Never remove the call in the large models (-m0 or -m1)
;-----;

;EXTERN ?SEG_INIT_L17
;LCALL ?SEG_INIT_L17 ; Initialize segments

;-----;
; If hardware must be initiated from assembly or if interrupts
; should be on when reaching main, this is the place to insert such code.
;-----;

    IF      single_chip

        MOV     R1,#0
        LCALL   main          ; main()
exit:      ; someone called exit()

    ELSE

        MOV     R6,#HIGH(xstack_end)
        MOV     R7,#LOW(xstack_end)

        IF      banked_mode

            EXTERN ?X_CALL_L18

            MOV     DPTR,#main
            LCALL   ?X_CALL_L18 ; main()
            DW      0           ; No parameter space
local_exit: ; someone called exit()

```

```
ELSE

MOV    DPTR,#0        ; No parameters

LCALL  main           ; main()
exit:   ; someone called exit()

ENDIF

ENDIF

;-----;
; Now when we are ready with our C program (usually 8051 C ;
; programs are continuous) we must perform a system-dependent ;
; action. In this simple case we just stop. ;
;-----;

SJMP   $              ; Forever...

IF     banked_mode

RSEG   FLIST

exit:  DD     local_exit

ENDIF

END    startup
```

```
-!                               -idaho.xcl-

      XLINK command file to be used with the 8051 C-compiler V2.xx
      using the -m4 to -m5 options (small)
      Usage: xlink your_file(s) -f idahoII

      First: define CPU  -!
-c8051

-!  Select register bank [0,8,10 or 18] -!
-D_R=0

-!  Setup all read-only segments (PROM) at zero -!
-Z(CODE)CSTART,RCODE,CODE,CDATA,CONST=0

-!  Setup all writable segments which must be mapped to internal RAM. -!
-Z(IDATA)DATA,UDATA,IDATA,ISTACK

-!  Load the 'C' library and assembly language functions -!
cl8051s
idaho
rdet1
event1
setgain
-x
-l idaho.map
-R
-FINTEL-STANDARD
-o idaho.hex
```


**Appendix B.4
Analysis Program Listing**

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
10      '
20      ' Program to read data from disk and plot PHA/RATIOS USING SA-10 DATA FORMAT.
30      '
35      ' Program Name: PROGSAM.BAS (INCLUDES AUTO GAIN NORMALIZATION)
40      '
70      ' REVISION 12/06/90 BY R. LAWRENCE
71      '
72      ' CHECKED 04/24/91 BY R.S.L.
73      '
74      ' REVISION 06/20/91 BY R. LAWRENCE
76      ' MODIFIED FOR SA-10 CARD READER, INCLUDING NEW DATA FORMAT USING
      ' PB-10 PROGRAMMER
77      '
78      ' REVISION 07/09/91 BY R. LAWRENCE
79      ' MODIFIED TO INCLUDE DOSE EQUIVALENT VALUES IN REM
80      '
81      ' REVISION 07/24/91 BY R. LAWRENCE
82      ' MODIFIED TO CORRECT GRAPHS FOR AMPLITUDE VS BLOCK NUMBER
83      '
84      ' REVISION 07/09/92 BY R. LAWRENCE
85      ' MODIFIED ENERGY CALCULATIONS AND DOSE
86      ' USE GNSTEP FOR FULL-SCALE ENERGY CALCULATIONS
88      '
90      '!!!! VALID ONLY FOR MARK-1 SYSTEM AND SA-10 OR PB-10 CARD READERS !!!!
92      '
94      '
96      '
98      ' set up dimensions and turn default key trapping off
100     '
110     OPTION BASE 1
120     DIM MX(9), MXCH(8)
142     DIM UNSH%(4, 8191), SHLD%(4, 8191)
160     DIM IR$(8), TEMP$(2), GNSTEP$(1001), A$(8), XYZ$(1001, 8), HVLT$(1001)
170     KEY OFF
171     KEES = 0
172     SCREEN 2
173     CLS
176     DATA 5.82749142,21.49667469,48.23436597,81.94909813,81.94909813,81.94909813,
81.94909813,81.94909813

177     FOR I = 1 TO 8: READ MAS$(I): NEXT I
180     KEES = 0
181     SERNO = 1
182     GOTO 610
183     '
```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
184      ***** SELECT SYSTEM AND FILENAME. *****
185      '
192      CLS
193      LOCATE 2, 1: PRINT "SELECT SYSTEM BOX. GRAY BOX = 1: SUITCASE = 2."
194      CLOSE
200      INPUT "", SYS
202      IF SYS = 2 THEN CLS : LOCATE 2, 1: PRINT "IF SPECIAL HESCO TEST DATA, ENTER 2.
ELSE ENTER 1.": INPUT "", SERNO

210      KEES = 1
300      IDUN% = 0
305      PHA = 0
310      PLAC% = 4
500      LOCATE PLAC%, 1: INPUT "Enter the name of the data file : ", FILE$
510      PLAC% = PLAC% + 1
520      OPEN FILE$ FOR RANDOM AS #1 LEN = 64
530      FIELD #1, 2 AS DELIM$, 4 AS DAT$, 10 AS TIM$, 16 AS IR1$, 16 AS IR2$, 12 AS TEMP$, 4
AS HVCONT$

535      RETURN
580      '
590      '      DEFINE FUNCTION KEYS AND DISPLAY KEY OPTIONS.
600      '
610      CLS
620      IDUN% = 1
640      LOCATE 24, 1: PRINT SPACE$(79);
650      LOCATE 25, 1: PRINT SPACE$(79);
660      ON KEY(1) GOSUB 850
670      ON KEY(2) GOSUB 3450
680      ON KEY(10) GOSUB 800
690      KEY(1) ON
700      KEY(2) ON
705      KEY(6) ON
710      KEY(10) ON
720      LOCATE 25, 1: PRINT "F1-DISP RATIO GRAPH";
730      LOCATE 25, 22: PRINT "F2-PRINT DATA";
740      LOCATE 25, 73: PRINT "F10-EXIT";
745      IF (PHA = 0) AND (KEES = 0) THEN ON KEY(3) GOSUB 7300: ON KEY(6) GOSUB 192:
KEY(3) ON: KEY(6) ON: LOCATE 25, 37: PRINT "F3-COMPUTE PHA F6-SELECT SYS/FILE";

746      IF (PHA = 0) AND (KEES = 1) THEN ON KEY(3) GOSUB 7300: KEY(3) ON: KEY(6) OFF:
LOCATE 25, 37: PRINT "F3-COMPUTE PHA";

748      IF (PHA = 1) AND (KEES = 0) THEN ON KEY(4) GOSUB 7540: ON KEY(6) GOSUB 192:
KEY(3) OFF: KEY(4) ON: KEY(6) ON: LOCATE 25, 37: PRINT "F4-SEL PHA CH. F6-SELECT
SYS/FILE";
```

```

750      '
760      ' wait for the user to decide what he wants to do (i.e. type a function key).
780      '
790      IF IDUN% = 1 THEN GOTO 660 ELSE GOTO 610
800      END
810      ' -----
820      ' ----- subroutines -----
830      ' -----
840      '
850      ' *****RATIO*****
860      '
870      '     SUBROUTINE TO GET INFORMATION FOR THE RATIO SUBROUTINE
880      '
890      CLS
910      PLAC% = 4
920      COMBIN% = 0
930      KIND% = 1
940      GOSUB 3690      ' see if we want combined data or not !
950      IF COMBIN% = 1 THEN GOTO 1270
960      '
970      ' *****REC*****
980      '
990      ' get record numbers to look at
1000     '
1010     LOCATE PLAC%, 1: PRINT "ENTER BLOCK NUMBER : ";
1020     INPUT "", REC
1030     IF (REC < 0) OR (REC > 16383) THEN GOTO 970
1040     PLAC% = PLAC% + 1
1210     '
1220     ' go get the requested data
1230     '
1240     GOSUB 4420
1242     IF GNSTEP# < 2! THEN GEV$ = "MeV": EGEV$ = "E-MeV": FSNRG# = (32 / (2 ^ (7 -
GNSTEP#))) * 1000: PWR# = 1000 ELSE GEV$ = "GeV": EGEV$ = "E-GeV": FSNRG# = (32 / (2 ^ (7 -
GNSTEP#))) * 1000: PWR# = 1

1245     FOR I = 1 TO 8: AVE#(I) = IR#(I): DTNRG#(I) = (IR#(I) / 4095#) * FSNRG#: NEXT I
1246     FOR I = 1 TO 8: DOSEQ#(I) = (1.60209# * (10 ^ (-5)) * DTNRG#(I)) / (PWR# * MAS#(I)):
NEXT I

1250     CLS
1270     PLAC% = 2
1280     '
1290     ' print the ratio data and then plot it to the screen
1300     '
1305     LOCATE PLAC%, 1: INPUT "SELECT SEPERATE (S) OR OVERLAY (O) GRAPHS."; OVRLY$

```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
1307  CLS
1310  GOSUB 1350
1320  GOSUB 1940
1321  ' LOCATE 4, 1: PRINT "IF NEW FILE ENTER 'N'. IF EXISTING FILE ENTER 'E'.": INPUT
" ", YYY$: CLS

1322  ' IF ((YYY$ = "N") OR (YYY$ = "n")) THEN GOTO 1325 ELSE GOTO 4133
1325  KEES = 0
1330  RETURN
1340  '
1350  ' *****PRINT RATIO DATA*****
1360  '
1370  ' SUBROUTINE TO PRINT THE RATIO DATA.
1380  '
1385  IF KIND% = 2 THEN GOTO 1472
1390  IF (COMBIN% = 0) AND ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC%, 1:
PRINT "BLOCK NO. "; REC; "WAS FIRED AT "; D$; " "; C$; ". "

1395  IF (COMBIN% = 0) AND ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN LOCATE 11, 1:
PRINT "FILENAME IS "; FILE$: LOCATE 12, 1: PRINT "BLOCK NO. "; REC; "WAS FIRED AT:":
LOCATE 13, 4: PRINT D$; " "; C$: LOCATE 14, 1: PRINT "FULL SCALE ENERGY IS:": _
LOCATE 15, 4: PRINT USING "###.##### \\"; FSNRG#; GEV$

1400  IF (COMBIN% = 1) AND ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC%, 1:
PRINT USING "START TIME = \ \ \ \ \. STOP TIME = \ \ \ \ \.": STD$; STC$; D$; C$

1402  IF (COMBIN% = 1) AND ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN LOCATE 11, 1:
PRINT "FILENAME IS "; FILE$: LOCATE 12, 1: PRINT "COMBINED "; S; "RECORDS:": LOCATE 13,
4: PRINT R; " THROUGH "; REC: LOCATE 14, 4: PRINT "START AT "; STD$; " "; STC$
: LOCATE 15, 4: PRINT "STOP AT "; D$; " "; C$: LOCATE 16, 1: PRINT "FULL SCALE ENERGY
IS:": LOCATE 17, 4: PRINT USING "###.##### \\"; MNNRG#; GEV$

1405  IF (COMBIN% = 0) AND ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC% - 1,
1: PRINT USING "FILENAME IS \ \. FULL SCALE ENERGY IS ###.##### \\"; FILE$; FSNRG#;
GEV$

1406  IF (COMBIN% = 1) AND ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC% - 1,
1: PRINT USING "FILE \ \ \ \ \; #### RECORDS:#### TO ####; ENERGY(FULL SCALE) = ###.#####
\ \.": FILE$; S; R; REC; MNNRG#; GEV$

1410  IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN PLAC% = PLAC% + 1 ELSE PLAC% =
PLAC% - 1

1430  LOCATE PLAC%, 1: PRINT "DETNO: D1 D2 D3 D4 D5 D6 D7
D8";
```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
1450 LOCATE PLAC% + 1, 1: PRINT "AVSIG:";
1460 FOR I = 1 TO 8: PRINT USING "#####.###"; AVE#(I); : NEXT I: PRINT
1470 IF KIND% <> 2 THEN GOTO 1600
1472 IF GNSTEP# < 2! THEN GEV$ = "MeV": EGEV$ = "E-MeV": FSNRG# = (32 / (2 ^ (7 -
GNSTEP#))) * 1000: PWR# = 1000 ELSE GEV$ = "GeV": EGEV$ = "E-GeV": FSNRG# = (32 / (2 ^ (7 -
GNSTEP#))) : PWR# = 1

1474 LPRINT USING "FILENAME IS \      \. FULL SCALE ENERGY IS ###.##### \. "; FILE$;
FSNRG#; GEV$

1475 LPRINT USING "BLOCK NO. ##### WAS FIRED AT \      \      \ "; REC; D$; C$
1480 LPRINT "          D1      D2      D3      D4      D5      D6      D7      D8"
1490 LPRINT "          -----"
1500 LPRINT "SIGNL:";
1510 FOR I = 1 TO 8: LPRINT USING "#####.###"; IR#(I); : NEXT I: LPRINT
1600 IF (IR#(4) = 0) THEN DT#(1) = 1 ELSE DT#(1) = IR#(4)
1605 IF (IR#(5) = 0) THEN DT#(2) = 1 ELSE DT#(2) = IR#(5)
1660 ,
1670 ' CALCULATE THE RATIO'S.
1680 ,
1690 FOR I = 1 TO 4
1700 RAT#(I) = (IR#(I) / DT#(1))
1710 RAT#(I + 4) = IR#(I + 4) / DT#(2)
1720 NEXT I
1725 IF KIND% = 2 THEN GOTO 1840
1730 LOCATE PLAC% + 2, 1: PRINT "NRMSG:";
1740 FOR I = 1 TO 8: PRINT USING "#####.###"; IR#(I); : NEXT I
1750 LOCATE PLAC% + 6, 1: PRINT "RATIO:";
1760 FOR I = 1 TO 8: PRINT USING "#####.###"; RAT#(I); : NEXT I: PRINT
1770 LOCATE PLAC% + 7, 1: PRINT "XVAL :";
1780 PRINT USING " #####.###"; .64; 1.229; 1.841; 2.4; .102; .508; 1.016; 1.524
1781 LOCATE PLAC% + 4, 1: PRINT USING "\      \ "; EGEV$;
1782 FOR I = 1 TO 8: PRINT USING "#####.###"; DTNRG#(I); : NEXT I
1783 LOCATE PLAC% + 5, 1: PRINT "DOSEQ:";
1784 FOR I = 1 TO 8: PRINT USING "##.##^"; DOSEQ#(I); : NEXT I
1785 IF COMBIN% <> 1 THEN GOTO 1790
1787 LOCATE PLAC% + 3, 1: PRINT "STDEV:";
1788 FOR I = 1 TO 8: PRINT USING "#####.###"; STDEV#(I); : NEXT I
1790 PLAC% = PLAC% + 7
1800 IF KIND% <> 2 THEN RETURN
1840 LPRINT "RATIO:";
1850 FOR I = 1 TO 8: LPRINT USING "#####.###"; RAT#(I); : NEXT I: LPRINT
1860 LPRINT "XVAL :";
1870 LPRINT USING " #####.###"; .64; 1.229; 1.841; 2.4; .102; .508; 1.016; 1.524
```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
1872  FOR I = 1 TO 8: DTNRG#(I) = (IR#(I) / 4095#) * FSNRG#: NEXT I
1873  FOR I = 1 TO 8: DOSEQ#(I) = (1.60209# * (10 ^ (-5)) * DTNRG#(I)) / (PWR# * MAS#(I)):
NEXT I

1874  LPRINT USING "\ \: "; EGEV$;
1876  FOR I = 1 TO 8: LPRINT USING "#####.###"; DTNRG#(I); : NEXT I
1877  LPRINT "DOSEQ:";
1878  FOR I = 1 TO 8: LPRINT USING "##.##^"; DOSEQ#(I); : NEXT I
1880  RETURN
1890  '
1900  ' *****CALL RATIO PLOTS*****
1910  '
1920  '   SUBROUTINE TO CALL PLOTS OF RATIO DATA
1930  '
1940  IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN TOP% = PLAC% + 1 ELSE TOP% =
PLAC% + 3

1950  TOPPIX% = (200 - ((25 - TOP%) * 8)) + 4
1960  GOSUB 2050
1970  GOSUB 2790
1980  IDUN% = 1
1990  LOCATE 25, 27: PRINT "type any key to continue";
2000  p$ = ""
2010  p$ = INKEY$
2020  IF p$ = "" THEN GOTO 2010
2025  LOCATE 25, 1: PRINT SPACE$(79);
2030  RETURN
2040  '
2050  ' *****PROCESS GRAPH OF UNSHIELDED DETECTORS*****
2060  '
2080  '
2090  ' find max and min ratio value that we will be plotting
2100  '
2110  MX = RAT#(1)
2120  IF RAT#(2) > MX THEN MX = RAT#(2)
2125  IF RAT#(3) > MX THEN MX = RAT#(3)
2130  IF RAT#(4) > MX THEN MX = RAT#(4)
2135  K = 1
2140  IF MX > 1 THEN K = 2
2145  IF MX > 2 THEN K = 5
2150  IF MX > 5 THEN K = 10
2160  IF MX > 10 THEN K = 20
2170  IF MX > 20 THEN K = 50
2180  IF MX > 50 THEN K = 100
2190  IF MX > 100 THEN K = 200
2200  IF MX > 200 THEN K = 500
```

```

2210 IF MX > 500 THEN K = 1000
2220 '
2230 ' set up some parameters for plotting and gosub to draw the
2240 ' basic graph.
2250 '
2260 IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN RIGHTPIX% = 300 ELSE RIGHTPIX% = 574
2270 CHARS = (LEN(STR$(K)) + 2)
2305 IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LEFTPIX% = 37 ELSE LEFTPIX% = 311
2320 TYP$ = "A"
2330 PLAC% = PLAC% + 2
2340 GOSUB 5170
2350 PLAC% = PLAC% - 2
2360 APOS#(1) = .64
2370 APOS#(2) = 1.229
2380 APOS#(3) = 1.841
2390 APOS#(4) = 2.4
2400 OFFSET = 0
2420 SLOPE# = 0!
2430 INTERCEPT# = 0!
2440 '
2450 ' plot the actual data now
2460 '
2470 FOR I = 1 TO 4
2480     X = LEFTPIX% + ((APOS#(I) / 2.667) * (RIGHTPIX% - LEFTPIX%))
2490     Y = 171 - (((RAT#(I + OFFSET)) / K) * (171 - TOPPIX%))
2500     LINE (X - 5, Y - 3)-(X + 5, Y + 3)
2510     LINE (X - 5, Y + 3)-(X + 5, Y - 3)
2520 NEXT I
2530 GOSUB 5920
2540 '
2550 ' plot the fit and write the slope and intercept
2560 ' to the screen
2570 '
2580 IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC% + 1, 6: PRINT USING
"SLOPE=##.####^ ^^ INT=##.####^ ^^"; SLOPE#, INTERCEPT#;

2581 IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) AND (COMBIN% = 0) THEN YY = 16: EXEX =
175: WYWY = 123

2582 IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) AND (COMBIN% = 1) THEN YY = 18: EXEX =
175: WYWY = 139

2583 IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN LOCATE YY, 1: PRINT "VARIABLE
DETECTORS: ( )": LOCATE YY + 1, 4: PRINT USING "SLOPE = ##.####^ ^^"; SLOPE#: LOCATE
YY + 2, 4: PRINT USING "INTERCEPT = ##.####^ ^^"; INTERCEPT#: LINE (EXEX - 5, WYWY _
+ 3)-(EXEX + 5, WYWY - 3): LINE (EXEX - 5, WYWY - 3)-(EXEX + 5, WYWY + 3)

```


MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
2584 GOSUB 2590
2585 RETURN
2586 '
2587 '*****DRAW SLOPE LINE*****
2588 '
2590 IF ((DSP$ = "LG") OR (DSP$ = "lg")) AND (TYP$ = "B") THEN GOTO 2605
2591 IF TYP$ = "A" THEN FRC1# = 2.667 ELSE FRC1# = 2!
2592 IF INTERCEPT# >= 0 AND INTERCEPT# <= K THEN Y1 = INTERCEPT#: X1 = 0:
GOTO 2595

2593 IF SLOPE# > 0 THEN Y1 = 0 ELSE Y1 = K
2594 X1 = (Y1 - INTERCEPT#) / SLOPE#
2595 Y2 = (SLOPE# * FRC1#) + INTERCEPT#
2596 IF Y2 >= 0 AND Y2 <= K THEN Y2 = Y2: X2 = FRC1#: GOTO 2599
2597 IF SLOPE# > 0 THEN Y2 = K ELSE Y2 = 0
2598 X2 = (Y2 - INTERCEPT#) / SLOPE#
2599 Y1 = ((1 - Y1 / K) * (172 - TOPPIX%)) + TOPPIX%
2600 Y2 = ((1 - Y2 / K) * (172 - TOPPIX%)) + TOPPIX%
2601 X1 = ((X1 / FRC1#) * (RIGHTPIX% - LEFTPIX%)) + LEFTPIX%
2602 X2 = ((X2 / FRC1#) * (RIGHTPIX% - LEFTPIX%)) + LEFTPIX%
2603 LINE (X1, Y1)-(X2, Y2)
2604 GOTO 2780
2605 LGINTR# = INTERCEPT#
2606 LGSLOP# = SLOPE#
2607 IF (LGINTR# >= MNLOG%) AND (LGINTR# <= MXLOG%) THEN Y(1) = LGINTR#: X(1)
= 0: GOTO 2609

2608 IF (LGINTR# > MXLOG%) THEN Y(1) = MXLOG%: X(1) = (-LGINTR# + MXLOG%) /
LGSLOP#

2609 Y(2) = (LGSLOP# * 2) + LGINTR#
2610 IF (Y(2) >= MNLOG%) AND (Y(2) <= MXLOG%) THEN Y(2) = Y(2): X(2) = 2: GOTO 2612
2611 IF (Y(2) < MNLOG%) THEN Y(2) = MNLOG%: X(2) = (LGINTR# - MNLOG%) / -LGSLOP#
2612 FOR I = 1 TO 2
2620 Y(I) = LN2 - (Y(I) * 1 / LGDEC) * YPIX
2622 X(I) = ((X(I) / 2) * (RIGHTPIX% - LEFTPIX%)) + LEFTPIX%
2623 NEXT I
2630 LINE (X(1), Y(1))-(X(2), Y(2))
2780 RETURN
2790 '
2800 '***** PROCESS GRAPH OF SHIELDED DETECTORS *****
2810 '
2830 '
2840 ' NOTE: No comments here since this subroutine is
2850 ' logically equivalent to the previous one.
2860 '
```

```

2870  MX# = RAT#(5)
2880  IF RAT#(6) > MX# THEN MX# = RAT#(6)
2885  IF RAT#(7) > MX# THEN MX# = RAT#(7)
2890  IF RAT#(8) > MX# THEN MX# = RAT#(8)
2891  MN# = RAT#(5)
2892  IF RAT#(6) < MN# THEN MN# = RAT#(6)
2893  IF RAT#(7) < MN# THEN MN# = RAT#(7)
2894  IF RAT#(8) < MN# THEN MN# = RAT#(8)
2895  K = 1: MXLOG% = 0
2900  IF MX# > 1 THEN K = 2: MXLOG% = 1
2903  IF MX# > 2 THEN K = 5
2905  IF MX# > 2.71828 THEN MXLOG% = 2
2910  IF MX# > 5 THEN K = 10
2915  IF MX# > 7.389056 THEN MXLOG% = 3
2920  IF MX# > 10 THEN K = 20
2930  IF MX# > 20 THEN K = 50
2935  IF MX# > 20.085536# THEN MXLOG% = 4
2940  IF MX# > 50 THEN K = 100
2945  IF MX# > 54.59815 THEN MXLOG% = 5
2950  IF MX# > 100 THEN K = 200
2955  IF MX# > 148.41359# THEN MXLOG% = 6
2960  IF MX# > 200 THEN K = 500
2965  IF MX# > 403.42879# THEN MXLOG% = 7
2970  IF MX# > 500 THEN K = 1000
2971  IF MN# = 1 THEN MNLOG% = 0
2972  IF (MN# < 1) AND (MN# >= .367879) THEN MNLOG% = -1
2973  IF (MN# < .367879) AND (MN# >= .13533528#) THEN MNLOG% = -2
2974  IF (MN# < .13533528#) AND (MN# >= .049787) THEN MNLOG% = -3
2975  IF (MN# < .049787) AND (MN# >= .018315639#) THEN MNLOG% = -4
2976  IF (MN# < .018315639#) AND (MN# >= .006737947#) THEN MNLOG% = -5
2977  LGDEC = MXLOG% - MNLOG%
2980  IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN RIGHTPIX% = 622 ELSE RIGHTPIX% = 574
2990  IF ((DSP$ = "LN") OR (DSP$ = "ln")) THEN CHARS = (LEN(STR$(K))+ 2) ELSE CHARS = 4
3025  IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LEFTPIX% = 359 ELSE LEFTPIX% = 311
3040  TYP$ = "B"
3050  PLAC% = PLAC% + 2
3060  GOSUB 5170
3070  PLAC% = PLAC% - 2
3080  APOS#(1) = .102
3090  APOS#(2) = .508
3100  APOS#(3) = 1.016
3110  APOS#(4) = 1.524
3120  OFFSET = 4
3140  SLOPE# = 0!
3150  INTERCEPT# = 0!

```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
3160  FOR I = 1 TO 4
3170      X = LEFTPIX% + ((APOS#(I) / 2!) * (RIGHTPIX% - LEFTPIX%))
3172      IF ((DSP$ = "LG") OR (DSP$ = "lg")) GOTO 3182
3180      Y = 171 - (((RAT#(I + OFFSET)) / K) * (171 - TOPPIX%))
3181      GOTO 3190
3182      LGEE# = LOG(RAT#(I + OFFSET))
3188      Y = LN2 - (LGEE# * 1 / LGDEC) * YPIX
3190      LINE (X - 5, Y)-(X + 5, Y): LINE (X, Y + 3)-(X, Y - 3)
3200      LINE (X + 5, Y + 3)-(X + 5, Y - 3): LINE (X + 5, Y + 3)-(X - 5, Y + 3): LINE (X - 5, Y
- 3)-(X - 5, Y + 3): LINE (X - 5, Y - 3)-(X + 5, Y - 3)

3210  NEXT I
3220  GOSUB 5920
3225  GOSUB 2590
3230  IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LOCATE PLAC% + 1, 46: PRINT USING
"SLOPE=##.###^"^ INT=##.###^"^"; SLOPE#, INTERCEPT#;

3231  IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) AND (COMBIN% = 0) THEN YY = 19: EXEX =
199: WYWY = 147

3232  IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) AND (COMBIN% = 1) THEN YY = 21: EXEX =
199: WYWY = 163

3234  IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN LOCATE YY, 1: PRINT "LEAD-SHIELD
DETECTORS: ( )": LOCATE YY + 1, 4: PRINT USING "SLOPE = ##.###^"^"; SLOPE#: LOCATE
YY + 2, 4: PRINT USING "INTERCEPT = ##.###^"^"; INTERCEPT#: LINE (EXEX - 5, _
WYWY)-(EXEX + 5, WYWY): LINE (EXEX, WYWY + 3)-(EXEX, WYWY - 3): LINE (EXEX - 5,
WYWY + 3)-(EXEX + 5, WYWY + 3)

3235  IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN LINE (EXEX - 5, WYWY - 3)-(EXEX + 5,
WYWY - 3): LINE (EXEX - 5, WYWY + 3)-(EXEX - 5, WYWY - 3): LINE (EXEX + 5, WYWY +
3)-(EXEX + 5, WYWY - 3)

3236  RETURN
3440  '
3450  '*****PRINT RATIO DATA WITH NO GRAPH *****
3460  '
3480  CLS
3501  LOCATE PLAC%, 1: PRINT "VERIFY PRINTER IS AT TOP OF PAPER FORM.";
3502  LOCATE PLAC% + 1, 1: PRINT "PRESS ENTER KEY WHEN READY.";
3504  INPUT FORM$
3506  CLS
3490  COMBIN% = 0
3500  KIND% = 2
3510  LOCATE PLAC%, 1: PRINT "ENTER STARTING RECORD NUMBER: ";
3520  INPUT "", REC1
```

```
3522 CLS
3524 LOCATE PLAC%, 1: PRINT "ENTER NUMBER OF RECORDS:";
3526 INPUT "", REC2
3528 CLS
3530 PLAC% = PLAC% + 1
3540 FOR REC = (REC1) TO (REC1 - 1 + REC2)
3570     CLS
3590     PLAC% = 4
3610     GOSUB 4420
3620     GOSUB 1350
3625     GOSUB 7000
3640 NEXT REC
3641 CLS
3650 IDUN% = 0
3655 KEES = 0
3660 RETURN
3680 '
3690 ' *****COMPUTE COMBINED DATA*****
3692 '
3700 CLS
3720 PLAC% = 4
3721 LOCATE PLAC%, 1: PRINT "SELECT LINEAR (LN) OR LOG (LG) DISPLAY FOR SHIELDED
DETECTORS."

3722 INPUT "", DSP$
3723 CLS
3730 LOCATE PLAC%, 1: PRINT "COMBINED DATA? [Y/N] : ";
3740 PLAC% = PLAC% + 1
3750 INPUT "", Z$
3760 IF Z$ = "N" OR Z$ = "n" THEN RETURN
3880 COMBIN% = 1
3890 CLS
3895 AVE# = 0
3900 FOR I = 1 TO 8: TOT#(I) = 0: NEXT I
3910 PLAC% = 4
3920 LOCATE PLAC%, 1: PRINT "ENTER STARTING BLOCK NUMBER : ";
3930 INPUT "", R
3940 LOCATE PLAC% + 1, 1: PRINT "ENTER NUMBER OF BLOCKS: ";
3950 INPUT "", S
4000 MNGNSP# = 7
4040 FOR I = 1 TO 8
4050     T#(I) = 0
4052     T2#(I) = 0
4054     TX2#(I) = 0
4060 NEXT I
4070 Q = 1
```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```

4080 REC = R - 1
4090 CLS
4110 REC = REC + 1
4112 PRINT Q
4115 GOSUB 4420
4117 IF REC = R THEN STD$ = D$: STC$ = C$
4121 FOR I = 1 TO 8
4122 IF IR#(I) >= 4094 THEN GOTO 4110
4123 NEXT I
4124 FOR I = 1 TO 8
4125 XYZ#(Q, I) = IR#(I)
4126 IF GNSTEP# < MNGNSP# THEN MNGNSP# = GNSTEP#
4127 GNSTEP#(Q) = GNSTEP#
4128 TOT#(I) = TOT#(I) + IR#(I)
4131 NEXT I
4132 IF Q < S THEN Q = Q + 1: GOTO 4110
4133 CLS : IF (SERNO = 2) THEN PRINT "ENTER AMPLIFIER GAIN.": BEEP: INPUT "",
AMPGN: CLS

4134 FOR X = 1 TO S
4138 FOR I = 1 TO 8
4142 IF ((SERNO = 2) AND (I = 8)) THEN XX = (1 / AMPGN) ELSE XX = 1
4144 A#(I) = XX * ((XYZ#(X, I)) * (2 ^ (GNSTEP#(X) - MNGNSP#)))
4146 T#(I) = T#(I) + A#(I)
4148 T2#(I) = T2#(I) + ((A#(I)) ^ 2)
4150 NEXT I
4155 NEXT X
4160 IF MNGNSP# < 2! THEN GEV$ = "MeV": EGEV$ = "E-MeV": MNNRG# = (32 / (2 ^ (7 -
MNGNSP#))) * 1000: PWR# = 1000 ELSE GEV$ = "GeV": EGEV$ = "E-GeV": MNNRG# = (32 / (2 ^ (7
- MNGNSP#))) * PWR# = 1

4161 FOR I = 1 TO 8
4162 TX2#(I) = ((T#(I)) ^ 2)
4180 IR#(I) = T#(I) / Q
4190 STDEV#(I) = SQR((T2#(I) - (TX2#(I) / Q)) / (Q - 1))
4196 DTNRG#(I) = (IR#(I) / 4095#) * MNNRG#
4197 DOSEQ#(I) = (1.60209# * (10 ^ (-5)) * DTNRG#(I)) / (PWR# * MAS#(I))
4198 AVE#(I) = TOT#(I) / Q
4200 NEXT I
4210 RETURN
4340 '
4350 ' ***** READ *****
4360 '
4370 ' subroutine to read in the proper record and convert the
4380 ' data to numeric format.
4390 '

```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
4400 ' read a record from the data file
4410 '
4420 GET #1, REC
4430 '
4440 ' do the date string
4450 '
4460 D$ = MID$(DAT$, 1, 2) + "-" + MID$(DAT$, 3, 2)
4470 '
4480 ' do the time string, taking care of the milliseconds field
4490 '
4500 C$ = MID$(TIM$, 1, 2) + ":" + MID$(TIM$, 3, 2) + ":" + MID$(TIM$, 5, 2)
4510 MSEC = VAL(MID$(TIM$, 7, 2)) * 10 + VAL(MID$(TIM$, 9, 1))
4520 MSEC$ = STR$(MSEC)
4530 MSEC$ = RIGHT$(MSEC$, LEN(MSEC$) - 1)
4540 C$ = C$ + "." + MSEC$
4550 '
4560 ' convert amplitude readings 4 at a time
4570 '
4580 ISTRT% = 1
4590 IEND% = 4
4600 TMP$ = IR1$
4610 JJ = 0
4620 FOR I = ISTRT% TO IEND%
4630     JJ = JJ + 1
4640     J = (JJ - 1) * 4 + 1
4650     A$ = MID$(TMP$, J, 2)
4660     GOSUB 4900
4670     IR#(I) = ANUM
4680     A$ = MID$(TMP$, J + 2, 2)
4690     GOSUB 4900
4700     IR#(I) = IR#(I) * 16 + ANUM / 16
4710 NEXT I
4720 IF ISTRT% <> 1 THEN GOTO 4780
4730 ISTRT% = 5
4740 IEND% = 8
4750 TMP$ = IR2$
4760 GOTO 4610
4770 '
4780 ' now do the temperatures
4790 '
4800 FOR I = 1 TO 3
4806     J = (I - 1) * 4 + 1
4809     A$ = MID$(TEMP$, J, 2)
4812     GOSUB 4900
4815     TMPR# = ANUM
4818     A$ = MID$(TEMP$, J + 2, 2)
```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
4821      GOSUB 4900
4822      TMPR# = TMPR# * 16 + ANUM / 16
4825      IF SYS = 2 THEN GOTO 4838
4826      IF I < 3 THEN TEMP#(I) = (TMPR# * 450 / 4095) - 273.15: GOTO 4865
4827      IF I = 3 THEN HIVOLT# = ((TMPR# * 4500 / 4095) + 10)
4828      IF HIVOLT# <= 420 THEN ABL# = .104038 - ((HIVOLT# - 390) * .0031095#)
4829      IF HIVOLT# > 420 AND HIVOLT# <= 460 THEN ABL# = .010753 - ((HIVOLT# - 420) *
.000947175#)

4830      IF HIVOLT# > 460 AND HIVOLT# <= 500 THEN ABL# = -.027134 - ((HIVOLT# - 460) *
.001120275#)

4831      IF HIVOLT# > 500 AND HIVOLT# <= 540 THEN ABL# = -.071945 + ((HIVOLT# - 500)
* .000264675#)

4832      IF HIVOLT# > 540 AND HIVOLT# <= 590 THEN ABL# = -.061358 + ((HIVOLT# - 540)
* .00062724#)

4833      IF HIVOLT# > 590 AND HIVOLT# <= 640 THEN ABL# = -.029996 + ((HIVOLT# - 590)
* .00090422#)

4834      IF HIVOLT# > 640 THEN ABL# = .015215 + ((HIVOLT# - 640) * .00090422#)
4835      FSNRG# = 10 ^ ((4.10872 - (HIVOLT# * .006924824#)) + ABL#)
4836      CLS
4837      GOTO 4865
4838      IF I < 3 THEN TEMP#(I) = (TMPR# * 482 / 4095) - 273.15
4839      IF I = 3 THEN HIVOLT# = TMPR# * 4820 / 4095
4840      A$ = MID$(HVCONT$, 1, 2)
4842      GOSUB 4900
4844      GNSTEP# = ANUM
4860      CLS
4865      NEXT I
4869      RETURN
4870      '
4880      '***** CONVERT ASCII TO DECIMAL *****
4890      '
4900      B$ = LEFT$(A$, 1)
4910      IF B$ > "/" AND B$ < ":" THEN ANUM = ASC(B$) - 48 ELSE ANUM = ASC(B$) - 55
4920      B$ = RIGHT$(A$, 1)
4930      T = ANUM
4940      IF B$ > "/" AND B$ < ":" THEN ANUM = ASC(B$) - 48 ELSE ANUM = ASC(B$) - 55
4950      ANUM = ANUM + T * 16
4960      RETURN
5150      '

```

MARK-1: RADIATION DETECTION SYSTEM

TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
5160 ' *****DRAW GRAPH *****
5170 '
5180 ' subroutine to draw the actual graph axes and tics and to label
5190 ' the whole mess.
5200 '
5210 LINE (LEFTPIX%, TOPPIX%)-(RIGHTPIX%, TOPPIX%)
5220 LINE (LEFTPIX%, TOPPIX%)-(LEFTPIX%, 172)
5230 LINE (LEFTPIX%, 172)-(RIGHTPIX%, 172)
5240 LINE (RIGHTPIX%, TOPPIX%)-(RIGHTPIX%, 172)
5250 XPIX = RIGHTPIX% - LEFTPIX%
5260 YPIX = 172 - TOPPIX%
5270 '
5280 ' draw the tic marks and label the major tics
5290 '
5300 XSTP = XPIX / 10
5310 X = LEFTPIX%
5320 FOR I = 0 TO 9
5330 '
5340 '     major tic so label it
5350 '
5360     X = LEFTPIX% + XSTP * I
5365     IF I MOD 3 <> 0 THEN R = 3 ELSE R = 5
5370     LINE (X, 172)-(X, (172 - R))
5380     LINE (X, TOPPIX%)-(X, (TOPPIX% + R))
5385     IF TYP$ = "A" THEN FRAC# = .8 ELSE FRAC# = .6
5390     IF I MOD 3 = 0 THEN VALU = ((I / 3) * FRAC#) ELSE GOTO 5490
5420     XPOS = (X * 8)
5430     IF XPOS MOD 64 <> 0 THEN RNDUP = 1 ELSE RNDUP = 0
5440     XPOS = XPOS / 64 + RNDUP
5442     IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) OR (((OVRLY$ = "O") OR (OVRLY$ = "o")))
AND (TYP$ = "A")) THEN H = 23 ELSE H = 11

5450     LOCATE H, XPOS - 2: PRINT USING "#.##"; VALU;
5490 NEXT I
5500 IF TYP$ = "A" THEN GOTO 5540
5510 XPOS% = (40 - CHARS - 21) / 2 + CHARS
5515 IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN WYE = 39 + XPOS% ELSE WYE = 34 +
XPOS%

5520 LOCATE 24, WYE: PRINT "LEAD SHIELD THICKNESS(cm)";
5530 IF TYP$ = "B" AND ((DSP$ = "LG") OR (DSP$ = "lg")) THEN GOTO 5680 ELSE GOTO 5560
5540 IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN EX = ((40 - CHARS - 18) / 2 + CHARS) - 2:
EY = 24 ELSE EX = 46: EY = 10

5550 LOCATE EY, EX: PRINT "DETECTOR THICKNESS(cm)";
5560 YSTP = YPIX / 10
```



```

5580   FOR I = 0 TO 9
5590       Y = 172 - (YSTP * I)
5600       IF I MOD 3 <> 0 THEN R = 4 ELSE R = 8
5610       LINE (LEFTPIX%, Y)-((LEFTPIX% + R), Y)
5620       IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) OR ((DSP$ = "LN") OR (DSP$ = "ln")) THEN
LINE (RIGHTPIX%, Y)-(RIGHTPIX% - R, Y)

5630       IF I MOD 3 = 0 THEN VALU = (I / 10) * K ELSE GOTO 5678
,
,   EMBEDDED SUBROUTINE
,

5650       L$ = STR$(VALU)
5660       L$ = RIGHT$(L$, LEN(L$) - 1)
5670       L = LEN(L$)
5671       IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) OR (TYP$ = "A") THEN XPOS% =
(LEFTPIX% * 8! / 64!) - L ELSE XPOS% = (RIGHTPIX% * 8! / 64!) + 5 - L

5672       IF (XPOS% * 8!) > LEFTPIX% THEN XPOS% = XPOS% - 1
5673       IF XPOS% < 1 THEN XPOS% = 1
5674       YPOS% = (Y + 4) / 8
5675       LOCATE YPOS%, XPOS%: PRINT L$;
5676       IF ((DSP$ = "LG") OR (DSP$ = "lg")) AND (TYP$ = "B") THEN RETURN
,
,   END EMBEDDED SUBROUTINE
,

5678   NEXT I
5679   GOTO 5790
5680   LNZ = 172 + (1 / LGDEC) * YPIX * MNLOG%
5681   IF ((MXLOG% > 0) AND (MXLOG% <= 2)) THEN GOTO 5682 ELSE GOTO 5700
5682   IF MXLOG% = 1 THEN MXI# = 2.5 ELSE MXI# = 7
5683   FOR I = 1 TO MXI# STEP (.5 * MXLOG%): GOSUB 5687: NEXT I
5684   IF MNLOG% = -1 THEN MNI# = .4 ELSE MNI# = .2
5685   FOR I = 1 TO MNI# STEP (-.2): GOSUB 5687: NEXT I
5686   GOTO 5790
,
,   EMBEDDED SUBROUTINE
,

5687   Y = LNZ - (LOG(I) * 1 / LGDEC) * YPIX: FLAGG = 0
5688   IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN LINE (LEFTPIX%, Y)-((LEFTPIX% + 4), Y)
5689   LINE (RIGHTPIX%, Y)-((RIGHTPIX% - 4), Y)
5690   IF ((MXLOG% > 0) AND (MNLOG% >= -2) AND (I MOD 1 = 0)) THEN VALU = I:
FLAGG = 1

5692   IF ((MXLOG% > 0) AND (MNLOG% <= -3) AND (I MOD 1 = 0)) THEN VALU = I:
FLAGG = 1

```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
5693     IF ((MXLOG% = 0) AND (MNLOG% = -1) AND (J% MOD 2 = 0)) THEN VALU = 1: FLAGG
= 1

5694     IF ((MXLOG% = 0) AND (MNLOG% < -1) AND (J% MOD 2 = 0)) THEN VALU = 1:
FLAGG = 1

5696     IF FLAGG = 1 THEN GOSUB 5650
5698     RETURN
,
'     END EMBEDDED SUBROUTINE
,

5700     IF MNLOG% = -1 THEN MNI% = 4 ELSE MNI% = 2
5710     FOR J% = 10 TO MNI% STEP (-1): I = J% / 10: GOSUB 5687: NEXT J%
5790     CHARS = YPIX * 25! / 200!
5800     IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN STRT = PLAC% + (CHARS - 5!) / 2! ELSE
STRT = PLAC% + ((CHARS - 5!) / 2!) + 2

5805     IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN WYE = 1 ELSE WYE = 35
5810     LOCATE STRT, WYE: PRINT "R";
5812     LOCATE STRT + .5, WYE: PRINT "A";
5814     LOCATE STRT + 1, WYE: PRINT "T";
5816     LOCATE STRT + 2.5, WYE: PRINT "I";
5818     LOCATE STRT + 3, WYE: PRINT "O";
5820     IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN WYE = 34 ELSE GOTO 5852
5822     LOCATE STRT - 1.5, WYE: PRINT "D";
5824     LOCATE STRT, WYE: PRINT "E";
5826     LOCATE STRT + .5, WYE: PRINT "T";
5828     LOCATE STRT + 1, WYE: PRINT "E";
5830     LOCATE STRT + 2.5, WYE: PRINT "C";
5832     LOCATE STRT + 3.5, WYE: PRINT "T";
5834     LOCATE STRT + 4.5, WYE: PRINT "O";
5836     LOCATE STRT + 5, WYE: PRINT "R";
5852     IF ((OVRLY$ = "S") OR (OVRLY$ = "s")) THEN WYE = 41 ELSE WYE = 78
5854     LOCATE STRT, WYE: PRINT "R";
5856     LOCATE STRT + .5, WYE: PRINT "A";
5858     LOCATE STRT + 1, WYE: PRINT "T";
5860     LOCATE STRT + 2.5, WYE: PRINT "I";
5862     LOCATE STRT + 3, WYE: PRINT "O";
5863     IF ((OVRLY$ = "O") OR (OVRLY$ = "o")) THEN WYE = 77 ELSE GOTO 5878
5864     LOCATE STRT - 3.5, WYE: PRINT "L";
5865     LOCATE STRT - 2, WYE: PRINT "E";
5866     LOCATE STRT - 1.5, WYE: PRINT "A";
5867     LOCATE STRT, WYE: PRINT "D";
5868     LOCATE STRT + 1, WYE: PRINT "S";
5869     LOCATE STRT + 2.5, WYE: PRINT "H";
5870     LOCATE STRT + 3, WYE: PRINT "I";
```

```

5871 LOCATE STRT + 4.5, WYE: PRINT "E";
5872 LOCATE STRT + 5, WYE: PRINT "L";
5873 LOCATE STRT + 6.5, WYE: PRINT "D";
5878 RETURN
5879 '
5880 ' ***** FIT *****
5890 '
5900 '     fitting subroutine
5910 '
5920 SY = 0
5930 SX = 0
5940 SX2 = 0
5950 SXY = 0
5980 FOR I = 1 TO 4
5985     IF ((DSP$ = "LG") OR (DSP$ = "lg")) AND (OFFSET = 4) THEN RATO#(I + OFFSET) =
LOG(RAT#(I + OFFSET)) ELSE RATO#(I + OFFSET) = RAT#(I + OFFSET)

5990     SX = SX + APOS#(I)
6000     SY = SY + RATO#(I + OFFSET)
6010     SX2 = SX2 + APOS#(I) * APOS#(I)
6020     SXY = SXY + APOS#(I) * RATO#(I + OFFSET)
6030 NEXT I
6040 INTERCEPT# = (SY * SX2 - SX * SXY) / (4 * SX2 - SX * SX)
6050 SLOPE# = (4 * SXY - SX * SY) / (4 * SX2 - SX * SX)
6060 RETURN
6070 '
6080 ' ***** PRINT SLOPE AND INTERCEPT - NO GRAPH *****
6090 '
7000 APOS#(1) = .64
7010 APOS#(2) = 1.229
7020 APOS#(3) = 1.841
7030 APOS#(4) = 2.4
7040 OFFSET = 0
7060 SLOPE# = 0!
7070 INTERCEPT# = 0!
7080 GOSUB 5920
7110 LPRINT "FOR DETECTORS #1 TO #4:";
7120 LPRINT USING " SLOPE = ###.###^ ^ ^ INTERCEPT = ###.###^ ^ ^"; SLOPE#;
INTERCEPT#;

7130 LPRINT
7140 APOS#(1) = .102
7150 APOS#(2) = .508
7160 APOS#(3) = 1.016
7170 APOS#(4) = 1.524
7180 OFFSET = 4

```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
7200 SLOPE# = 0!
7210 INTERCEPT# = 0!
7220 GOSUB 5920
7250 LPRINT "FOR DETECTORS #5 TO #8:";
7260 LPRINT USING " SLOPE = ###.###^ ^ INTERCEPT = ###.###^ ^"; SLOPE#;
INTERCEPT#: LPRINT : LPRINT

7270 RETURN
7300 '
7310 ' ***** COMPUTE PHA *****
7320 '
7330 CLS
7331 COMBIN% = 1
7332 PHA = 1
7333 PLAC% = 2
7334 LOCATE PLAC% + 1, 1: PRINT "SELECT PHA PLOT (P) OR AMPLITUDE VS. BLOCK
NUMBER PLOT (A).": INPUT "", PORA$: CLS

7370 LOCATE PLAC% + 1, 1: PRINT "ENTER STARTING BLOCK NUMBER:";
7380 INPUT "", R
7390 LOCATE PLAC% + 2, 1: PRINT "ENTER NUMBER OF BLOCKS:";
7400 INPUT "", S
7401 MNGNSP# = 7
7402 ERASE UNSH%, SHLD%, MX, MXCH
7404 FOR I = 1 TO 8: LOW(I) = 4094: HIGH(I) = 0: MXLVL#(I) = 0: MNLVL#(I) = 4094: NEXT I
7405 MNHV# = 4000: MXHV# = 0
7418 Q = 1
7422 REC = R - 1
7426 CLS
7430 REC = REC + 1
7434 GOSUB 4420
7435 IF REC = R THEN STD$ = D$: STC$ = C$
7436 IF ((PORA$ = "A") OR (PORA$ = "a")) THEN GOTO 7458
7438 FOR I = 1 TO 8
7442 IF (((PORA$ = "P") OR (PORA$ = "p")) AND (IR#(I) >= 4094)) THEN GOTO 7430
7446 NEXT I
7450 IF GNSTEP# < MNGNSP# THEN MNGNSP# = GNSTEP#
7458 FOR I = 1 TO 8: XYZ#(Q, I) = IR#(I): NEXT I
7468 GNSTEP#(Q) = GNSTEP#: HVLT#(Q) = HIVOLT#
7469 IF Q < (S + 1) THEN Q = Q + 1: GOTO 7430
7470 BEEP: LOCATE PLAC% + 1, 1: PRINT "IF PHA HAS BEEN SELECTED, THEN SELECT RAW
OR NORMALIZED DATE (R OR N).": INPUT "", NORML$: CLS

7471 FOR X = 1 TO S
7472 FOR J = 1 TO 8
7473 IF (((PORA$ = "P") OR (PORA$ = "p")) AND ((NORML$ = "N") OR (NORML$ = "n")))
```

```

THEN XYZ#(X, J) = XYZ#(X, J) * (2 ^ (GNSTEP#(X) - MNGNSP#))

7474     IF XYZ#(X, J) > MXLVL#(J) THEN MXLVL#(J) = XYZ#(X, J)
7475     IF XYZ#(X, J) < MNLVL#(J) THEN MNLVL#(J) = XYZ#(X, J)
7476     IF ((PORA$ = "A") OR (PORA$ = "a")) AND (HVL#(X) > MXHV#) THEN MXHV# =
HVL#(X) ELSE MXHV# = MXHV#

7477     IF ((PORA$ = "A") OR (PORA$ = "a")) AND (HVL#(X) < MNHV#) THEN MNHV# =
HVL#(X) ELSE MNHV# = MNHV#

7479     NEXT J
7480     NEXT X
7481     IF ((PORA$ = "A") OR (PORA$ = "a")) THEN GOTO 7525
7482     IF MNGNSP# < 2! THEN GEV$ = "MeV": EGEV$ = "E-MeV": MNNRG# = (32 / (2 ^ (7 -
MNGNSP#))) * 1000 ELSE GEV$ = "GeV": EGEV$ = "E-GeV": MNNRG# = (32 / (2 ^ (7 - MNGNSP#)))

7483     FOR X = 1 TO S
7484     FOR J = 1 TO 4
7486         IF (MXLVL#(J) > 4093) THEN DTNRG#(J) = 2 * MNNRG# ELSE DTNRG#(J) = MNNRG#
7487         IF (MXLVL#(J + 4) > 4093) THEN DTNRG#(J + 4) = 2 * MNNRG# ELSE DTNRG#(J +
4) = MNNRG#

7488         IF (MXLVL#(J) > 4093) THEN A% = XYZ#(X, J) / 2 ELSE A% = XYZ#(X, J)
7489         IF (MXLVL#(J + 4) > 4093) THEN B% = XYZ#(X, J + 4) / 2 ELSE B% = XYZ#(X, J + 4)
7492         IF A% = 0 THEN A% = 1
7493         IF B% = 0 THEN B% = 1
7500         UNSH%(J, A%) = UNSH%(J, A%) + 1
7502         SHLD%(J, B%) = SHLD%(J, B%) + 1
7510         IF UNSH%(J, A%) > MX(J) THEN MX(J) = UNSH%(J, A%): MXCH(J) = A%
7512         IF SHLD%(J, B%) > MX(J + 4) THEN MX(J + 4) = SHLD%(J, B%): MXCH(J + 4) = B%
7513         IF (UNSH%(J, A%) > 0) AND (A% < LOW(J)) THEN LOW(J) = A%
7514         IF (UNSH%(J, A%) > 0) AND (A% > HIGH(J)) THEN HIGH(J) = A%
7515         IF (SHLD%(J, B%) > 0) AND (B% < LOW(J + 4)) THEN LOW(J + 4) = B%
7516         IF (SHLD%(J, B%) > 0) AND (B% > HIGH(J + 4)) THEN HIGH(J + 4) = B%
7518     NEXT J
7520     NEXT X
7525     CLS
7526     GOSUB 7540
7527     KEES = 0
7532     RETURN
7534     '
7535     '***** PRINT PHA *****
7536     '
7540     PLAC% = 2
7541     CLS
7543     LOCATE PLAC%, 1: PRINT "ENTER DETECTOR NUMBER (1 TO 8) FOR PHA OR

```

MARK-1: RADIATION DETECTION SYSTEM

TECHNICAL SPECIFICATIONS MANUAL

Appendices

AMPLITUDE VS. BLOCK NUMBER:";

```
7544 LOCATE PLAC% + 2, 30: PRINT "**** OR ****"
7545 LOCATE PLAC% + 4, 1: PRINT "ENTER '9' FOR HIGH VOLTAGE VS. BLOCK NUMBER:"
7548 INPUT "", L
7549 CLS
7551 LOCATE PLAC%, 1: PRINT "SELECT AUTOMATIC HORIZONTAL SCALING (A) OR
FULL-SCALE (F) OR EXPANDED (E).";

7552 INPUT "", X$
7553 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "E") OR (X$ = "e")) THEN CLS :
LOCATE PLAC%, 1: PRINT "ENTER LOWEST PHA CHANNEL TO BE PLOTTED.": INPUT "",
MANLO(L): CLS : LOCATE PLAC%, 1: PRINT "ENTER HIGHEST PHA CHANNEL TO BE
PLOTTED.": INPUT "", MANHI(L): IF MANHI(L) MOD 25 = 0 THEN MANHIPH = MANHI(L) - 1
ELSE MANHIPH = MANHI(L)

7554 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "E") OR (X$ = "e")) THEN MULT =
((((MANHIPH \ 25) + 1) * 25) - ((MANLO(L) \ 25) * 25)) / 25: MLSM = (MANLO(L) \ 25) * 25:
GOTO 7574

7556 IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "E") OR (X$ = "e")) THEN CLS :
LOCATE PLAC%, 1: PRINT "ENTER LOWEST BLOCK NUMBER TO BE PLOTTED.": INPUT "",
BLKLO(L): CLS : LOCATE PLAC%, 1: PRINT "ENTER HIGHEST BLOCK NUMBER TO BE
PLOTTED.": INPUT "", BLKHI(L): IF BLKHI(L) MOD 25 = 0 THEN BLKHIGH = BLKHI(L) - 1 ELSE
BLKHIGH = BLKHI(L)

7558 IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "E") OR (X$ = "e")) THEN MULT =
((((BLKHIGH \ 25) + 1) * 25) - ((BLKLO(L) \ 25) * 25)) / 25: MLSM = (BLKLO(L) \ 25) * 25:
GOTO 7574

7560 IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "A") OR (X$ = "a")) THEN IF (R + S)
MOD 25 = 0 THEN SUM = (R + S - 1) ELSE SUM = R + S

7562 IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "A") OR (X$ = "a")) THEN MULT =
((((SUM \ 25) + 1) * 25) - ((R \ 25) * 25)) / 25: MLSM = ((R \ 25) * 25): GOTO 7574

7564 IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "F") OR (X$ = "f")) THEN MULT = 800:
MLSM = 0: GOTO 7574

7566 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "F") OR (X$ = "f")) THEN MULT = 200
ELSE MULT = (((HIGH(L) \ 25) + 1) * 25) - ((LOW(L) \ 25) * 25) / 25

7570 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "F") OR (X$ = "f")) THEN MLSM = 0
ELSE MLSM = (LOW(L) \ 25) * 25

7574 CLS
```

MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
7575 TOPPIX% = 16
7576 BOTPIX% = 136
7578 LEFTPIX% = 100
7580 RIGHTPIX% = 580
7590 LINE (LEFTPIX%, (TOPPIX% - 4))-(RIGHTPIX%, (TOPPIX% - 4))
7600 LINE (LEFTPIX%, (TOPPIX% - 4))-(LEFTPIX%, (BOTPIX% - 4))
7610 LINE (LEFTPIX%, (BOTPIX% - 4))-(RIGHTPIX%, (BOTPIX% - 4))
7620 LINE (RIGHTPIX%, (TOPPIX% - 4))-(RIGHTPIX%, (BOTPIX% - 4))
7630 XSTP = (RIGHTPIX% - LEFTPIX%) / 25
7640 FOR I = 0 TO 25
7650     X = LEFTPIX% + XSTP * I
7660     IF I MOD 5 <> 0 THEN Q = 3 ELSE Q = 5
7670     LINE (X, (BOTPIX% - 4))-(X, (BOTPIX% - 4 - Q))
7680     LINE (X, (TOPPIX% - 4))-(X, (TOPPIX% - 4 + Q))
7690     IF I MOD 5 = 0 THEN VALU = ((I * MULT) + MLSM) ELSE GOTO 7740
7700     XPOS = X / 8
7722     IF I = 0 THEN XPOS = XPOS - 2
7730     LOCATE 18, XPOS - 2: PRINT USING "#####"; VALU
7740 NEXT I
7750 LOCATE 19, 36: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "CHANNEL NUMBER"
ELSE PRINT "BLOCK NUMBER"

7755 IF ((PORA$ = "A") OR (PORA$ = "a")) THEN GOTO 7822
7760 K = 10
7770 IF MX(L) > 10 THEN K = 20
7780 IF MX(L) > 20 THEN K = 50
7790 IF MX(L) > 50 THEN K = 100
7800 IF MX(L) > 100 THEN K = 200
7810 IF MX(L) > 200 THEN K = 500
7820 IF MX(L) > 500 THEN K = 1000
7821 GOTO 7830
7822 IF L = 9 THEN MAX# = MXHV#: MIN# = MNHV# ELSE MAX# = MXLVL#(L): MIN# =
MNLVL#(L)

7824 MLTV = (((((MAX# \ 25) + 1) * 25) - ((MIN# \ 25) * 25)) / 5000)
7825 MLSV = (MIN# \ 25) * 25
7830 YSTP = (TOPPIX% - BOTPIX%) / 10
7840 FOR I = 0 TO 10
7850     Y = BOTPIX% + (YSTP * I)
7860     IF ((I MOD 2 = 0) AND ((PORA$ = "P") OR (PORA$ = "p"))) THEN VALU = (K / 10) * I
ELSE IF ((I MOD 2 = 0) AND ((PORA$ = "A") OR (PORA$ = "a"))) THEN VALU = ((I * 500 * MLTV)
+ MLSV) ELSE GOTO 7880

7870     LOCATE (Y / 8), 9: PRINT USING "#####"; VALU
7880     LINE (LEFTPIX%, (Y - 4))-(LEFTPIX% + 10, (Y - 4))
7890     LINE (RIGHTPIX%, (Y - 4))-(RIGHTPIX% - 10, (Y - 4))
```

MARK-1: RADIATION DETECTION SYSTEM TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
7900  NEXT I
7902  LOCATE 4, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) OR (((PORA$ = "A") OR (PORA$ =
"a")) AND (L < 9)) THEN PRINT "" ELSE PRINT "H"

7904  LOCATE 5, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) OR (((PORA$ = "A") OR (PORA$ =
"a")) AND (L < 9)) THEN PRINT "" ELSE PRINT "I"

7905  LOCATE 6, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "A" ELSE IF L = 9 THEN PRINT "G"

7910  LOCATE 7, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "C" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "M" ELSE IF L = 9 THEN PRINT "H"

7920  LOCATE 8, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "O" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "P" ELSE IF L = 9 THEN PRINT ""

7930  LOCATE 9, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "U" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "L" ELSE IF L = 9 THEN PRINT "V"

7940  LOCATE 10, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "N" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "I" ELSE IF L = 9 THEN PRINT "O"

7950  LOCATE 11, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "T" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "T" ELSE IF L = 9 THEN PRINT "L"

7952  LOCATE 12, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "S" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "U" ELSE IF L = 9 THEN PRINT "T"

7953  LOCATE 13, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "D" ELSE IF L = 9 THEN PRINT "A"

7954  LOCATE 14, 7: IF ((PORA$ = "P") OR (PORA$ = "p")) THEN PRINT "" ELSE IF ((PORA$ =
"A") OR (PORA$ = "a")) AND (L < 9) THEN PRINT "E" ELSE IF L = 9 THEN PRINT "G"

7955  LOCATE 15, 7: IF L = 9 THEN PRINT "E"
7958  IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "E") OR (X$ = "e")) THEN MANLO(L) =
MANLO(L): MANHI(L) = MANHI(L) ELSE IF ((PORA$ = "P") OR (PORA$ = "p")) THEN MANLO(L)
= LOW(L): MANHI(L) = HIGH(L) ELSE IF ((PORA$ = "A") OR (PORA$ = "a")) AND
((X$ = "E") OR (X$ = "e")) THEN MANLO(L) = BLKLO(L): MANHI(L) = BLKHI(L) ELSE IF ((PORA$
= "A") OR (PORA$ = "a")) THEN MANLO(L) = R: MANHI(L) = R + S - 1

7961  IF ((PORA$ = "P") OR (PORA$ = "p") OR (PORA$ = "A") OR (PORA$ = "a")) AND ((X$ =
"F") OR (X$ = "f")) THEN D = 1 ELSE D = (MANLO(L) \ 25) * 25
```


MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL

Appendices

```
7962 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "F") OR (X$ = "f")) THEN E = 4094
ELSE IF ((PORA$ = "A") OR (PORA$ = "a")) AND ((X$ = "F") OR (X$ = "f")) THEN E = 16382 ELSE
IF MANHI(L) MOD 25 = 0 THEN E = (((MANHI(L) - 1) \ 25) + 1) * 25 ELSE _
E = ((MANHI(L) \ 25) + 1) * 25

7970 FOR B = MANLO(L) TO (MANHI(L) - 1)
7980 FOR I = 1 TO 2
7981 IF B = 0 THEN C = B + I ELSE C = B + I - 1
7982 IF ((PORA$ = "A") OR (PORA$ = "a")) THEN GOTO 7997
7986 IF L > 4 THEN GOTO 7994
7990 YPOS(I) = (1 - ((BOTPIX% - TOPPIX%) * (UNSH%(L, C) / K))) + (BOTPIX% - 4)
7992 GOTO 8000
7994 YPOS(I) = (1 - ((BOTPIX% - TOPPIX%) * (SHLD%((L - 4), C) / K))) + (BOTPIX% - 4)
7995 GOTO 8000
7997 IF L < 9 THEN YPOS(I) = -((BOTPIX% - TOPPIX%) * ((XYZ#(C - R + 1, L) - MLSV) /
(((MXLVL#(L) \ 25) + 1) * 25) - MLSV))) + BOTPIX% - 4

7998 IF L = 9 THEN YPOS(I) = -((BOTPIX% - TOPPIX%) * ((HVLTH(C - R + 1) - MLSV) /
(((MAX# \ 25) + 1) * 25) - MLSV))) + BOTPIX% - 4

8000 IF ((PORA$ = "P") OR (PORA$ = "p")) AND ((X$ = "F") OR (X$ = "f")) THEN G = (C /
5000) ELSE G = ((C - D) / (E - D))

8005 XPOS(I) = ((RIGHTPIX% - LEFTPIX%) * (G)) + (LEFTPIX%)
8010 NEXT I
8020 LINE (XPOS(1), YPOS(1))-(XPOS(2), YPOS(2))
8030 NEXT B
8032 IF ((PORA$ = "A") OR (PORA$ = "a")) THEN GOTO 8082
8040 LOCATE 20, 7: PRINT USING "FILENAME IS \ \. FULL SCALE ENERGY IS ###.##### \
\."; FILE$; DTNRG#(L); GEV$

8050 LOCATE 21, 7: PRINT USING "COMBINED #### RECORDS: ##### THROUGH #####."; S; R;
REC

8055 LOCATE 22, 7: PRINT USING "START TIME = \ \ \ \. STOP TIME = \ \ \ \.";
STD$; STC$; D$; C$

8060 LOCATE 23, 7: PRINT USING "MAXIMUM COUNTS IN DETECTOR NUMBER # ARE #### IN
CHANNEL NUMBER #####."; L; MX(L); MXCH(L)

8080 LOCATE 24, 7: PRINT USING "LOWEST/HIGHEST CHANNELS WITH DATA ARE #### /
####"; LOW(L); HIGH(L);

8081 GOTO 9000
```

**MARK-1: RADIATION DETECTION SYSTEM
TECHNICAL SPECIFICATIONS MANUAL**

Appendices

8082 LOCATE 20, 7: IF (L < 9) THEN PRINT USING "FILENAME IS \ \. DETECTOR NUMBER
IS # ."; FILE\$; L ELSE PRINT "FILENAME IS "; FILE\$; "."

8083 LOCATE 21, 7: IF (L < 9) THEN PRINT USING "AMPLITUDE VS. BLOCK NUMBER:
BLOCKS ##### TO #####."; MANLO(L); MANHI(L) ELSE PRINT USING "HIGH VOLTAGE VS. BLOCK
NUMBER: BLOCKS ##### TO #####."; MANLO(L); MANHI(L)

8084 LOCATE 22, 7: PRINT USING "START TIME = \ \ \ \. (BLOCK NUMBER #####)";
STD\$; STC\$; R

8085 LOCATE 23, 7: PRINT USING "STOP TIME = \ \ \ \. (BLOCK NUMBER #####)"; D\$;
C\$; R + S - 1

9000 RETURN

END

**DATE
FILMED**

8/1/71 93