THE REMOTE WORKING LEVEL MONITOR

Prepared for
UNITED STATES DEPARTMENT OF THE INTERIOR
BUREAU OF MINES

by
Argonne National Laboratory
with the
U. S. Energy Research and Development Administration
Argonne, Illinois

FINAL REPORT
on
The Remote Working Level Monitor
Contract No. EO 252053
Design and Construction of a Remote Working Level Meter

18 November 1977
DISCLAIMER NOTICE

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or recommendations of the Interior Department's Bureau of Mines or of the U. S. Government.
Abstract

The Remote Working Level Monitor (RWLM) is an instrument used to remotely monitor the Rn-daughter concentrations and the Working Level (WL). It is an ac powered, microprocessor based instrument which multiplexes two independent detector units to a single central processor unit (CPU). The CPU controls the actuation of the detector units and processes and outputs the data received from these remote detector units. The remote detector units are fully automated and require no manual operation once they are set up. They detect and separate the alpha emitters of RaA and RaC as well as detecting the beta emitters of RaB and RaC. The resultant pulses from these detected radioisotopes are transmitted to the CPU for processing. The programmed microprocessor performs the mathematical manipulations necessary to output accurate Rn-daughter concentrations and the WL. A special subroutine within the program enables the RWLM to run and output a calibration procedure on command. The data resulting from this request can then be processed in a separate program on most computers capable of BASIC programming. The calibration program results in the derivation of coefficients and beta efficiencies which provides calibrated coefficients and beta efficiencies.

Key Words
Remote Working Level Monitor (RWLM)
Working Level (WL)
Rn-daughter Concentrations
FOREWORD

This report was prepared by Argonne National Laboratory, Electronics Division, 9700 S. Cass Avenue, Argonne, Illinois 60439 under Contract Number HO 252053. The contract was initiated under the Metal and Nonmetal Health and Safety Research Program. It was administered under the technical direction of DMRC with Mr. Robert Drouillard acting as the Technical Project Officer. Mr. David Askin was the contract administrator for the Bureau of Mines.

This report is a summary of the work recently completed as part of this contract during the period April 1975 to June 1977. This report was submitted by the authors on June 1977.
Subject Inventions

The design of the filter-advance mechanism was recently submitted to the Chicago Patent Group of the Energy Research and Development Administration for a patent review. The Chicago Patent Group of ERDA will contact the Bureau of Mines regarding the disposition of this patent application in accordance with the contract.
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>RWLM System</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2</td>
<td>RWLM Detector Unit #1 - Covers in place</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3</td>
<td>RWLM Detector Unit #2 - Covers removed</td>
<td>10</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Central Processing Unit (CPU)</td>
<td>11</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Exposed View of Central Processing Unit (CPU)</td>
<td>12</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Schematic of Filter Transport Mechanism</td>
<td>13</td>
</tr>
</tbody>
</table>

1.0 Introduction                                            14

2.0 Theory                                                  16

3.0 Descriptions                                            19

3.1 The Detector Assembly                                    19

3.1.1 The mechanical filter-transport assembly              19

3.1.2 The air-pump and motor-telemeter drive               21

3.1.3 Regulated DC power supplies                           21

3.1.4 Radiation detectors                                   22

3.1.5 Detector preamplifier and single-channel analyzer     23

3.1.6 The electronic control package                        24

3.1.6.1 Line drivers and opto-isolated line receivers      24

3.1.6.2 Tachometer output window comparator, high-voltage regulator, solenoid driver and pump-motor speed control circuit 25

3.1.6.3 Slo-Syn motor translator control                   27
3.1.6.4 Source check paper position and paper empty logic

3.2 Central Processing Unit (CPU)

3.2.1 IMSAI MPU-A theory of operation
3.2.2 IMSAI RAM 4-1 theory of operation
3.2.3 Cromenco 8K Bytesaver read only memory
3.2.4 Processor technology 2KRO 2K read only memory
3.2.5 Processor technology 3P+S input output module
3.2.6 Vectored interrupt board
3.2.7 Head control on select board
3.2.8 Line receiver board
3.2.9 The accumulator board
3.2.10 Front panel replacement card
3.2.11 System bus
3.2.12 CPU power supply - functional description

3.3 Control Program

3.3.1 Program description
3.3.2 The language
3.3.3 Calibration coefficients

3.4 Floating Point Math Pack

3.4.1 Floating point routines
3.4.2 Floating point accumulator

3.5 Format Conversion Package

3.6 Interface between PL/M and Assembly Language

Floating Point Math Pack Routines
4.0 Calibration

4.1 Measurement of Flowrate V

4.2 Determinations of $E_1$, $E_2$, and $E_3$

4.3 Discriminator Adjustments

4.4 Calibration Program

Figure 7 Data Terminal Dialog between Operator and System

Figure 8 RWLM Calibration Program

Figure 9 RWLM Calibration Program - Readout with calculate

(C) beta efficiency branch

Figure 10 RWLM Calibration Program - Readout with input (I)

beta efficiency branch

Figure 11 Sample Calibration Data Readout

4.5 Calibration Procedure

5.0 Test Results

Table 5.1

6.0 Operating Instructions

6.1 Normal Operation

6.1.1 Filter loading procedures

6.1.2 Cable hook-up

6.1.3 Power

6.1.4 Remote power

6.1.5 Power check

6.1.6 Terminal and restart

6.1.7 Reset operation

6.1.8 Set date and time

6.1.9 Set altitude correction factor

6.1.10 Normal operation
### 6.2 Source Check Mode

6.2.1 Mechanical procedure  
6.2.2 Operation of source check

### 6.3 Calibration Mode

### 7.0 References

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A</td>
<td>90</td>
</tr>
<tr>
<td>Appendix B</td>
<td>90</td>
</tr>
<tr>
<td>Appendix C</td>
<td>91</td>
</tr>
<tr>
<td>References</td>
<td>92</td>
</tr>
</tbody>
</table>
Figure 2  E/WLH Detector Unit #1 - Covers in place
Figure 3  HVLH Detector Unit #2 - Covers removed
Figure 4 Central Processing Unit (CPU)
Figure 5 Exposed View of Central Processing Unit (CPU)
Figure 6 Schematic of Filter Transport Mechanism
1.0 INTRODUCTION

The exact methods commonly used to determine Working Level (WL) measure the total alpha activity on a filter membrane during three preselected time periods. Due to the half-lives of the isotopes involved (TRaB = 26.8 min. TRaC = 19.7 min.), these methods require 35 minutes to complete. Methods of measuring WL using the total alpha counts, where the counting time is less than about one RaB half-life after the end of sampling, are unsatisfactory because of statistical problems. In addition, these methods are almost blind to RaB (Pb\(^{214}\)) as a consequence of its relatively long half-life and the fact that RaB is a beta emitter. The solution to this dilemma is to measure RaB directly through its beta activity. Since RaB's daughter RaC (Bi\(^{214}\)) is also a beta emitter, the two beta spectra have to be separated to obtain RaB. How this is accomplished is described in subsequent sections describing the theory of the Remote Working Level Monitor/Instant Working Level Meter (RWLM/IWLM) and its calibration.

It is important to note that the IWLM, described in the Bureau of Mines Final Report prepared under contract No. HO 122106, and the RWLM described here use the same method to obtain the measurement of the radon-daughter concentrations and the WL. The IWLM is a portable battery-operated instrument which automatically performs the sequence of operations required after initiation of the start button. Its output is displayed visually and must be manually recorded. The RWLM is a multichannel (two channel in this case) totally automatic instrument designed to measure working levels and individual Rn-daughter concentrations and record the data. The RWLM is capable of taking up to 200 samples at operator-selected time intervals. The capability
to perform these automatic operations is obtained through the use of a microprocessor as the instrument controller. The microprocessor enables the RWLM to keep real time, provide timing logic for the electronic sequencing necessary for proper operation, perform the necessary mathematical operations, communicate with the outside world through the data terminal, and perform its three different modes of operation. The microprocessor, in addition to performing the circuit logic through the use of software, also provides an automatic operational check of four mechanical functions and a functional check on remote power. It checks for proper operation of the filter positioning, adequate filter supply, release of the solenoid-operated air seal brake and monitors the pump speed within preselected limits. If any one of these parameters are beyond limits or not operational, the computer is flagged, resulting in a vectored interrupt which provides the specific error message to be printed on the data terminal and stops the measurement sequence. A Texas Instrument's Silent 700 data terminal is used for data recording and instrument control.

The instrument is thoroughly described in section 2. Its theory of operation and calibration are described in sections 2 and 4. Test results are tabulated in section 5, and section 6 contains the operating procedures for its three modes of operation. The drawings and computer programs are contained in Appendices A, B, and C.
2.0 THEORY

After taking a background count for two minutes, the RWLM collects an air sample for two minutes onto a filter membrane, moves the filter membrane to a counting station and then counts this air sample for two minutes starting 13 sec after the end of the sampling period. The instrument has three counting channels, the lower-energy alpha channel, the upper-energy alpha channel, and a beta sensitive channel. The RaA counts observed are accumulated in the lower-energy alpha channel; the RaC' counts are stored in the upper-energy alpha channel while the total beta counts from RaB and RaC are recorded in the beta channel. The counts in these three channels are functions of $N_A$, $N_B$, $N_C$, the unknown concentrations of RaA, RaB, and RaC in the ambient air (units are atoms/liter). The following relationships hold:

$$
TA = 0.558814 \ E_1 V N_A
$$

$$
B + C = (0.038498E_2 + 0.001793E_2)VN_A + (0.097712E_2 + 0.007473E_1)VN_B + 0.130234E_3 VN_C
$$

$$
TC' = (0.001793N_A + 0.007473N_B + 0.130234N_C)E_1 V
$$

(1)

where:

$TA$ = alpha counts from RaA accumulated during the 2-min counting period starting 13 sec after the end of sampling.

$B + C$ = beta counts from RaB and RaC in beta channel accumulated during the same time interval as above.

$TC'$ = alpha counts from RaC' (same period of accumulation as above).

$V$ = flowrate (liters/min).
The numerical coefficients in (1) follow from the laws of radioactive-series decay. The half-lives used are:

<table>
<thead>
<tr>
<th>Nuclide</th>
<th>Half-life</th>
</tr>
</thead>
<tbody>
<tr>
<td>RaA(^{210})Po</td>
<td>3.05 min</td>
</tr>
<tr>
<td>RaB(^{214})Pb</td>
<td>26.8 min</td>
</tr>
<tr>
<td>RaC(^{214})Bi</td>
<td>19.7 min</td>
</tr>
</tbody>
</table>

To show the principle of calculation of these numerical coefficients, a sample calculation is given below:

**EXAMPLE:** Calculate the numerical coefficient for the first of (1).

This coefficient is the product of three factors:

1) **Buildup factor** = \(1 - \exp(-\lambda_A t_B)\)
   
   \(t_B = \text{buildup time} = 2 \text{ min.}\)

2) **Delay factor** = \(\exp(-\lambda_A D)\)
   
   \(D = \text{Delay before start of the counting interval} = 13 \text{ sec or } 13/60 \text{ min.}\)

3) **Decay factor** = \(\left[1 - \exp(-\lambda_A t_D)\right] / \lambda_A\)
   
   \(t_D = \text{decay time} = 2 \text{ min.}\)
   
   \(\lambda_A = \text{RaA decay constant} = 0.2272614.\)
Multiplying all these factors one obtains:

\[ 1 - \exp \left( -\frac{\lambda}{A} t_B \right) \exp \left( -\frac{\lambda}{A} t_D \right) \left( 1 - \exp \left( -\frac{\lambda}{A} t_D \right) \right) \frac{1}{\lambda_A} = 0.558814. \]

The derivations of the analogous coefficients for the other equations are more complex but easily obtainable by the computer programs given in Appendix C. If all the efficiencies and the flowrate are known, (1) contains only numerical coefficients and the unknowns \( N_A, N_B, \) and \( N_C. \) Inverting (1) results in another set of equations which give \( N_A', N_B', \) and \( N_C' \) for every observed set of \( T_A, (B+C), \) and \( T_C'. \) The WL can be calculated easily from the resulting \( N_A', N_B', \) and \( N_C'. \)

The RWLM method does not make any assumptions about the equilibrium conditions between the Rn daughters, but does assume that the airborne concentrations \( N_A, N_B, \) and \( N_C \) remain constant during the sampling period of two minutes. A detailed description of how the efficiencies are obtained and how (1) is inverted is given in section 4.0 and in the computer programs in Appendix C.
3.0 DESCRIPTIONS

The Remote Working Level Monitor (RWLM) consists of two independent detector assemblies, a central processing unit and a data terminal. (See Figs. 1 through 6.)

3.1 The Detector Assembly

Each detector assembly (see Figs. 2 and 3) is housed in a portable aluminum instrument case. This high-density package contains:

1. The mechanical drive mechanisms for the filter-transport assembly.
2. The air pump and its motor-tachometer assembly.
3. The necessary power supplies for the mechanical and electronic subassemblies.
5. The electronic preamplifiers, discriminators, and single-channel analyzer.
6. The electronic control package.

3.1.1 The mechanical filter-transport assembly (see Fig. 6) advances the filter membrane 2.000 ± 0.005 inches each transport cycle. The sequence of mechanical operation is as follows:

1. The filter paper is advanced two cycles to provide a clean surface for background measurements.
2. The pump draws a sample.
3. The filter paper is advanced again to move the filtered sample to the detectors for counting.
Six inches of filter membrane* are used per measurement. Each sensor has a sufficient filter supply to allow up to 200 samples, either one sample per hour for eight days or any other programmed frequency limited by a total of 200 samples and a 14 min or greater sampling period.

This operation is motor driven, and computer controlled. As shown in Fig. 6, a 5V, bifilar-wound stepping motor** provides 200 steps per revolution. A 5:1-90° drive and reduction gear couples the motor to a "Scotch Yoke" drive assembly. One revolution of this drive assembly will provide one filter-advance cycle and requires 1000 pulses from the stepping motor per cycle.

The filter advance cycle is implemented as follows: The power to the magnetic clutch is turned off, releasing the clutch which is under mechanical spring pressure. Upon activation of pulses to the digital motor, the filter advance roller moves down due to the mechanical conversion of angular to linear motion of the scotch yoke mechanism. This advances the paper from the storage reel. At the 500th pulse the filter advance roller is at its bottom dead center position, which completes the filter advance. At this time, the computer energizes the magnetic clutch.

* GELMAN ACROPOR AN-800
** Superior Electronics SLO-SYN Model M061-FC02.
braking the filter membrane and preventing further motion of the filter. Due to the action of the slip clutch and the one-way bearing on the take-up reel and its drive, the filter paper is taken up during the last 500 pulses of the motor operation. The four-phase drive signal to the stepping motor is generated by a digital motor-translator circuit explained in section 3.1.6.3.

3.1.2 The Air-Pump and Motor-Tachometer Drive

A GAST rotary-vane vacuum pump, Model 1033, is coupled with a universal self-aligning coupler to a printed circuit motor-tachometer combination, Model U9M4T. The pump-motor system is servo-regulated using tachometer feedback for a constant-speed operation. It is activated under computer control. The interface circuit is described in section 3.1.6.3. The pump speed is adjusted to provide a flowrate of 12 liters per min. The computer-controlled solenoid-operated magnetic clutch assures a tight air seal of the filter against the inlet air port.

3.1.3 Regulated DC Power Supplies

The following regulated DC power supplies are included within each portable detector assembly:

1. An ELEKTON Model OLV30-15 (15V @ 3.3 amp) regulated power supply provides power for the air pump motor.
2. An Analog Devices Model PS 933 (±24V @ 50 mA) regulated power supply and a ±12V @ 50 mA regulated supply (drawing KL-C-7171) provide power for the alpha detector and preamp.

3. A Semiconductor Circuits, Model MP 1.5. 750/2.15.100 (±15V @ 100 mA plus +5V @ 750 mA) regulated power supply provides power for the linear circuits and the digital logic.

4. A Semiconductor Circuits, Model DPS 1.5.1500 (+5V @ 1.5 A) regulated power supply provides power for the solenoid.

5. A 5V @ 3 A regulated supply (see drawing KL-C-7171, sheet 7, Appendix B) provides power for the stepping motor.

The AC line power is supplied to the detector assemblies via a power cable from the central processing unit and is independently fused within the detector assembly. Each detector assembly also contains its own power "ON/OFF" switch.

### 3.1.4 Radiation Detectors

The alpha detector is a silicon-surface barrier detector.* The face of the detector is protected with double-aluminised Mylar (25 gauge). The detector is sensitive to both the 6.00 MeV RaA and the 7.68 MeV RaC′ alpha particles.

*Ortec #CA-29-300-100.
The beta detector is a NE 102 Scintillator optically coupled to a 10-stage, low-noise, high-gain photomultiplier (EMI 9633B). The gamma-background sensitivity has been reduced by a factor of 50 over that of the original MIT MLM by the provision of an integral lead shield for the beta detector and the use of a thin scintillator (0.003 in).

3.1.5 Detector Preamplifier and Single-Channel Analyzer

The alpha channel utilizes a Canberra preamplifier Model 1406, whose output is further amplified by a dual integrated-circuit amplifier having a gain of 100. This combination nets a charge gain of 20 V/pico-coulomb for the alpha channel. The output of this charge-sensitive amplifier is delivered to a single-channel analyzer which separates the 6.00 MeV RaA energy peaks from the 7.68 MeV RaC' energy peaks. (See Appendix C, drawing EL-C-7171, sheet 12.) These separated energy pulses are routed via line drivers to the appropriate accumulators in the central processing unit (CPU).

The beta channel’s preamplifier is contained within the photomultiplier tube, voltage divider and preamplifier housing. The charge gain of this amplifier is fixed at .05 V/pico-coulomb. A regulated high-voltage converter (Venue K30) is used to supply a well regulated voltage to the dynode bias divider to insure the gain stability of the photomultiplier.
A voltage of approximately 1250V was found to be a satisfactory value for the PM tube. At this value the optimum signal-to-noise ratio is obtained. The output pulses from the PM tubes' charge-sensitive amplifier are routed to a low-level discriminator which prevents the counting of noise pulses. The signals that satisfy the discriminator requirements are shaped, routed to a line driver and then to the beta-channel accumulator in the CPU.

3.1.6 The Electronic Control Package

This package contains four printed circuit cards which perform the following functions:

Card 1: Line drivers and opto-isolated line receivers
(see drawing EL-C-7171, sheet 10 of Appendix B).

Card 2: Tachometer output window comparator, high-voltage regulator, solenoid driver and the pump-motor speed control (see drawing EL-C-7171, sheet 11 of Appendix B).

Card 3: Slo-Syn motor translator control (see drawing EL-C-7171, sheet 4 of Appendix B).

Card 4: Source check logic control and paper position and paper empty comparators (see drawing EL-C-7171, sheet 9 of Appendix B).

3.1.6.1 Line Drivers and Opto-Isolated Line Receivers

The two detector assemblies and the central processing unit are separated by up to 500 ft of shielded cable. To drive digital pulses,
with minimum distortion over this distance, requires line drivers and line receivers. The line drivers are designed to drive a 50 ohm line which is terminated in the characteristic impedance of the cable. Opto-isolators are used to terminate the line, as well as to provide electrical isolation between the three units. This further reduces noise-induced signals and electrically isolates the computer from the detection heads. Signals from the CPU are terminated within the particular detector assembly and the return signal is returned to the CPU ground via the remaining wire of the twisted pair. Signals from the detector assembly are terminated within the CPU and the ground is returned in a similar fashion to the detector assembly.

3.1.6.2 Tachometer Output Window Comparator, High-Voltage Regulator, Solenoid Driver and Pump-Motor Speed Control Circuit

Since the pump speed is approximately proportional to the flowrate, the subject of course to wear the efficiency, pump speed is regulated using a tachometer. (See Appendix B, sheet 6, tachometer vs flowrate graph.) The tachometer voltage is sensed
and electrically compared to an upper and lower limit. If the tachometer voltage does not fall within this preselected window, it is sensed as an error by the comparator. The comparator output is monitored by the CPU and an off normal signal causes the error message "FLOWRATE OUT-OF-RANGE" to be types on the terminal. If this occurs, the measurement cycle is aborted and the CPU returns to command loop.

A buffered reference signal is fed to the low-voltage input of a high-voltage DC to DC converter (see drawing EL-C-7171, sheet 11). A temperature compensated zener reference feeds a potentiometer which provides a variable reference for a closed-loop feedback-buffer amplifier. This amplifier supplies the regulated voltage and current necessary for the regulated high-voltage converter.

This regulated high voltage is used by the beta channel's high-voltage photomultiplier bias string. Regulating this voltage provides a measure of stability in the gain of the photomultiplier.

The solenoid driver provides the necessary current to activate the solenoid-driven clutch.
The circuit itself is basically a current-controlled switch. A provision for a solenoid status check signal is fed back to the computer which checks for this signal at the appropriate times. If this flag signal is present when checked, the message "SOLENOID NOT RELEASED" is printed on the terminal and the measurement operation is aborted.

A closed-loop servo-controlled motor-drive circuit is provided to regulate the angular velocity of the pump motor, keeping the pump at a constant regulated speed. Pump activation is computer controlled.

3.1.6.3 Slo-Syn Motor Translator Control
A four phase, 5V, one Amp, step sequence (full-step mode) is required to drive the stepping motor. This four-bit driver code is generated by the translator control card. This circuit operates as follows: A hexadecimal divider is either counted up or down depending on the motor-rotation direction desired. The binary output of this divider provides a four-bit address to a programmable read-only memory which has the proper motor code stored. The memory outputs provide the current drivers with the proper input code.
sequence. The chip-enable input of the PROM provides an On/Off switch for these drivers. The motor direction is controlled by either an up or down count, the motor stepping speed is controlled by the pulse-repetition rate to the counter, and power is provided by the PROM chip-enable input. This motor can be driven to a maximum speed of 330 steps/sec of a 3 sec paper-advance cycle.

3.1.6.4 Source Check Paper Position and Paper Empty Logic

In the source-check mode of operation, the discriminator setting for the RaC' energy must be lowered in order to detect the Am^{241} source used in the source check. To accomplish this, a linear switch is automatically actuated by the CPU to provide a lower voltage to the upper-level channel when a C' source check operation is selected. The upper-level discriminator is usually set at approximately 1.3 volts to detect the 7.6 MeV RaC' energy. When a source check mode is entered, this voltage is automatically set to approximately 0.4V.

The paper position and paper empty comparators provide logic signals to the CPU which enable the CPU to check these parameters. To
accomplish this, each comparator receives a signal from a light-sensitive detector (Skan-a-matic). These detectors provide their own infrared light source and are able to detect reflected IR energy and provide a current output proportional to the magnitude of IR-reflected light received. The current output is fed through a high-impedance resistor which provides a signal voltage to the comparators where it is compared to a common reference voltage. As long as the light is not reflected, the current switch is turned off. However, when this light source is detected, due to reflection, the current switch is turned on until the paper is exhausted. When the filter feed reel is empty, the light source projects against a black surface causing no reflection. The paper-position detector looks for a white scribe mark etched on one end of the scotch yoke cam drive assembly which appears within the view of the detector only when the filter advance roller is at top dead center.

3.2 Central Processing Unit (CPU)

The RWLM central processing unit consists of a chassis with an integral power unit containing the following:
1) an IMSAI MPU-A-8080 processor board
2) an IMSAI RAM 4-1 random access memory board with 1K of memory.
3) a Cromemco 8K read only memory board
4) a Processor Technology 2KRO -2K read only memory board
5) a Processor Technology 3P+S - serial I/O board
6) a vectored interrupt board
7) a head control and line driver board
8) a line receiver board
9) an accumulator board
10) a front panel substitute board
11) a system bus board
12) a system power supply

The CPU is housed in an IMSAI corporation main frame modified to meet the needs of the RWLM. The modification consists of the addition of system I/O cable connectors and system power connectors.

3.2.1 IMSAI MPU-A Theory of Operation

The IMSAI MPU-A board is structured around the Intel 8080A microprocessor chip, and much of the MPU-A board is wired to support the 8080A device. The MPU-A board provides interfacing between the 8080A chip and the data and address busses, clock and synchronization signals, and the voltage regulation necessary for the 8080A and other chips.
The address lines from the 8080A drive the address bus on the back plane through 8T97 tri-state buffer drivers. These drivers may be disabled through the ADDRESS DISABLE line on pin 22 of the back plane. Intel 8216 bi-directional bus drivers connect the 8080's bi-directional data bus to the back plane's dual uni-directional DATA IN and DATA OUT busses. The direction of data transmission is determined by the DIRECTION ENABLE line. The DIRECTION ENABLE line is in turn controlled by the front panel and the processor status signals DATA BUS IN and HALF ACKNOWLEDGE. The 8216 can be disabled by the DATA OUT DISABLE line on pin 23 of the back plane.

The 8080A's bi-directional data bus is also connected to the data bus socket and the 8212 status byte latch. The data bus socket is used to connect a front panel (not included) to the bi-directional bus, while the 8312 latch transfers the status byte to the back plane via 8T97 drivers. These drivers are disabled by the STATUS DISABLE line on pin 18 of the back plane. The 8212 is latched up by the STATUS STROBE signal of the 8224 clock chip to store the status information for each instruction cycle.

One K pullup resistors to +5 volts are connected to all the bi-directional bus lines to ensure that during the time the bus is not driven, the 8080A reads all 1's.

The 8824 clock ship and crystal oscillator, provide the two-phase non-overlapping 2 megacycle system clock for
the 8080A. These clocks are also driven onto the back plane through 8T97 tri-state buffered drivers.

The CLOCK line on the back plane is driven from the TTL Phase II clock line through a delay so that the phase relation of the clock signal to the Phase II and Phase I back plane signals is nearly identical to that produced by the MITS Altair 8800 system. Six sections of a 7404 are used for this delay to provide greater simplicity and higher reliability than a one shot. The 8224 chip also provides the power-on reset function through use of a 4.7K resistor and 33 uF capacitor connected to the reset input of the 8224. The power-on reset is applied to the 8080A and is applied to the POWER ON CLEAR line, pin 99 on the back plane.

The two BACK PLANE READY signals are ANDed and connected to the 8224 for synchronization with the Phase II clock before being connected to the 8080A chip. The INTERRUPT line is connected directly to the 8080A, while the HOLD REQUEST line is synchronized with the Phase II clock and then connected to the 8080A.

The six processor status signals (SYNC WRITE, STROBE DATA BIT IN, READ STROBE, INTERRUPT ENABLED, HOLD ACKNOWLEDGE, and WAIT ACKNOWLEDGE) are all driven onto the back plane through 8T97 tri-state buffered drivers. These drivers may be disabled by the CONTROL DISABLE line, pin 19 on the back plane.
The +4% volts is regulated from the +8 volts by a 7805 integrated circuit regulator, while the -5 volts is regulated by a 5-volt zener and a 470 ohm resistor from the 16-volt bus. The +12 volts is regulated by a 12-volt zener and connected to the +16-volt line by two 82 ohm 1/2 watt resistors in parallel. All voltages are filtered with .33 microfarad tantalum and disc ceramic capacitors.

3.2.2 IMSAI RAM 4-1 Theory of Operation

The RAM-4 board has space for 4K bytes of memory which consist of 32 chips of Intel 8111 or 2111 type random access memory organized 456 words x 4 bits wide in each chip. In the RWLM system only 1K bytes are implemented.

These RAM devices are arranged on the board in a 2 x N (1 ≤ N ≤ 16) array, with the top row A containing bits 0, 1, 2, 3 of all the data and Row B containing bits 4, 5, 6, and 7 of all the data. Read/write and address control is provided by a support network of Gates (C8, C9, C13) and a decoder (C10). Bi-directional tri-state bus drivers (C15, C16) are used to receive and transmit data to and from the IMSAI 8080 System bus.

To begin the Read or Write Cycles, the board must be enabled. As shown in the schematic, the board enable is produced by an 8-input NAND (741L830 in position C13). Four of the NAND inputs are the jumper selected board address bits (A12, A13, A14, A15 or complements), and the remaining two are the inverted status bits SINF and SOUT. When the board is properly addressed, the NAND
output is driven low. The 8205 1-of-8 decoder is then enabled, addressing a particular memory chip pair uniquely determined by the states of \(A_8, A_9, A_{10}\) and \(A_{11}\).

The 8T97 bus driver (C14) is also driven by the NAND (C13). Also enabled at this time are the 8216 (C15, C16) tri-state bi-directional bus drivers.

The direction of data flow is determined by the 7402 in position C8, which when low selects a data path going from the IMSAI 8080 data bus to the RAM-4 board's data bus. This is made low by either the memory write line from the control panel or the complement of the memory read status signal from the processor. Thus for normal operation, with the machine running, the status signal memory read determines whether these data bus drivers are driving to the IMSAI 8080 data-out bus. When the machine is stopped and the front panel is being used, the direction of data transmission is selected by the memory write pulse from the front panel.

When writing from the front panel, a delay is necessary before turning off the data on the memory chips (so that there is time for the memory chips to write on the trailing edge of the write strobe before the data disappears) and this delay is provided by the disc capacitor to ground connected at the output of the inverter at C9 pin 8. In addition to selecting the direction of data flow through the bi-directional data bus drivers, the direction control signal is also inverted and applied to the output disable pin on the 8111's so that during writing the 8111 is
receiving data on its bi-directional data pins and not attempting to drive. The write strobe is applied to the 8111's through a 4 section data out DIP switch which enables the programmer to turn off the write pulse for each K for debugging purposes. When the machine is running normally, the write comes from the processor write strobe line (pin 77 on the back plane) and when the front panel is being used, the write strobe line comes from the front panel on the memory write line (pin 60 on the back plane). Two other sections of the 7402 are used to take either one of these write strobes and buffer them to drive the memory chips.

3.2.3 **Cromenco 8K Bytesaver Read Only Memory**

This board is a read only memory board plug compatible with the standard S-100 microcomputer bus. It has the capacity to hold eight 2708 type UV erasable read only memories.

The board also contains provisions for programming the 2708 type memories; however, the RWLM does not support the software required to perform this function. In the RWLM seven of the eight sockets are used. This board is addressed at Zero and Stores the main RWLM program.

3.2.4 **Processor Technology 2KRO 2K Read Only Memory**

This board is a read only memory, socket compatible with the standard S-100 microcomputer bus. It has the capacity
to hold eight 1702A type UV erasable read only memories. Each 1702A is capable of holding 256 bytes of program. In the RWLM this board holds the floating point mathematics package and the floating point-PLM transfer routines.

3.2.5 Processor Technology 3P+S Input Output Module

This board is designed to provide the interface between the microcomputer system and the terminal device. In the RWLM this board is wired to implement the standard RS-232C signals required by the TI Silent 700.

The board contains a serial I/O port which links the terminal to the system and a parallel port which is designed to provide I/O status information to the processor. It also contains two parallel I/O ports that are not used by the RWLM.

3.2.6 Vectored Interrupt Board

The vectored interrupt board provides the processor with an eight level priority interrupt capability and a controlled interval clock that can be used to interrupt the processor on a regular basis. In the RWLM a one second interrupt is used to provide the timing information. The program control of the vector interrupt board is performed via output port OFBH. The output of the address select gate (74L30) is ANDed with processor signals PWR and SOUT. These signals are used to latch the lower four bits of data into the 8214 priority interrupt chip. When 8214 is enabled and one of its priority request lines is low, the 8314 INT line is
used to strobe an 8212 IC. This causes the 8212's INT
flip-flop to be set and requests an interrupt from the
processor. When the processor acknowledges the interrupt
with an INTA, the 8212 outputs are enabled and this
puts the interrupt request address on bits 3, 4, and 5
of the DATA IN bus. The remaining bits of DATA IN are
held high. The byte thus formed on the DATA IN lines
is a restart instruction with bits 3, 4, and 5 directing
the processor to one of eight restart locations.

The clock circuit consists of a set of frequency dividers
which count the phase 2 clock pulses. In the RWLM, the
one second intervals are selected. The clock interrupt
flip-flop is reset by ANDing the RST 7 instruction with
the INTA signal. Details of operation on the 8212 and the
8214 can be found in the Intel Data Book.

3.2.7 Head Control on Select Board

The Head control board consists of an I/O address decoder,
an 8212 used as an output port latch, and two sets of
line drivers. When a command is to be sent to the remote
heads an OUT 7 instruction is placed on the bus. This is
decoded by the 8205 whose output, along with PWR and SOUT
cause the 8212 to latch the output instruction. The output
lines of 8212 are then used to select the head and to hold
the required information. Bit 0 selects the head to be
addressed via a set of 7408 AND gates. When Bit 0 is equal
to 0, head one is selected and when Bit 0 is equal to one,
head two is selected. Bits 2 and 3 are not used. Bit 4 delivers pulses to the digital motor, Bit 5 turns on the digital motor, Bit 6 activates the solenoid and Bit 7 activates the pump.

3.2.8 **Line Receiver Board**

The line receiver board consists of a set of opto-isolator line receivers which detect the signals sent by each remote head. Head selection is performed by 72LS158 multiplexers which select either head one or head two using the head select bit from the output port on the Head control board.

This data is placed on the processors data bus whenever an IN 7 instruction is performed. Address decoding is performed on the Head Control Board and signals are bussed to the 8212 on the line receiver board.

3.2.9 **The Accumulator Board**

In order to count the pulses in each of the detection channels (RaA, Ra(B+C) and RaC'). A seven-decade accumulator has been implemented for each channel. The accumulator section consists of a MC 14518 decade prescaler which holds the least significant decade with a Mostek MK 50395N six decade up/down counter with latch and comparator.

The accumulator board is controlled as output port seven by an 8212 latch. The data is read out of the accumulators onto the data bus by a set of 8212 latches used as input ports, four, five and six. Output port 4's output commands are:
1) STOREL - This strobes the output 8212 and causes them to hold the data being impressed on their input lines.

2) SET - This resets the multiplexed output of the MK50395N to the most significant digit.

3) SCAN - This strobes the MK50395N output selector to output the next digit.

4) CLEAR - Resets all counters.

5) COUNT ENABLE - opens the count gates and allows the accumulators to collect data.

6) STORES - Enables latch in MK50395N.

The MK50395N is a N-MOS device and requires a VCC of +12 volts; therefore, open collector 7406 devices are used to interface with the output port and the RCA 3081. Seven transistor common emitter IC devices are used to interface with the input port.

3.2.10 Front Panel Replacement Card

In the normal S-100 system, a front panel is used which generates a MWRITE signal. When no front panel is installed, a way of generating this signal must be provided. A single 7400 NAND gate is used to implement the function.

MWRITE = PWR·SOUT

3.2.11 System Bus

The S100 system bus structure consists of 100 lines. These are arranged 50 on each side of the plug-in board, with pins 1 through 50 on the component side and pins 51 through 100 on the back side.
As the board is viewed right-side up (components up, 100 pin connector towards you) pin #1 is on the left end on the top and pin 51 is on the back side directly opposite pin #1.

Conventions:

SYMBOLS: "P" prefix indicates a processor command or control signal
"S" prefix indicates a processor status signal

LOADING: All inputs to a card should be loaded with a maximum of 1 TTL low power load

LEVELS: All bus signals except the power supply are TTL. All Data and Address lines are positive TRUE (ground = 0 bit)

BUS DEFINITION

<table>
<thead>
<tr>
<th>Front Side</th>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>+8V</td>
<td>+8 volts</td>
<td>Unregulated input to 5 V regulators</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>+16V</td>
<td>+16 volts</td>
<td>Positive unregulated voltage</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>XRDY</td>
<td>External Ready</td>
<td>Used by Front Panel: Pulling this line low will cause the processor to enter a WAIT state and allows the status of the normal Ready Line (PRDY) to be examined.</td>
</tr>
</tbody>
</table>
### BUS DEFINITION

#### Front Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>V10</td>
<td>Vectored Interrupt Line #0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>V11</td>
<td>Vectored Interrupt Line #1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>V12</td>
<td>Vectored Interrupt Line #2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>V13</td>
<td>Vectored Interrupt Line #3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>V14</td>
<td>Vectored Interrupt Line #4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>V15</td>
<td>Vectored Interrupt Line #5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>V16</td>
<td>Vectored Interrupt Line #6</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>V17</td>
<td>Vectored Interrupt Line #7</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>HS</td>
<td>Head Select Bit</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>SINP·PDBIN for Head Output Port</td>
<td></td>
</tr>
<tr>
<td>14-16</td>
<td>UNUSED</td>
<td>HEAD ADDRESS SELECT I/O SEL</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>STATUS DSBL</td>
<td>STATUS DISABLE</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>Allows the buffers for the 8 status lines to be tristated</td>
<td></td>
</tr>
</tbody>
</table>
**BUS DEFINITION**

**Front Side**

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>CC DSB</td>
<td>COMMAND CONTROL</td>
<td>Allows the buffers for the 6 output command/control lines to be tri-stated</td>
</tr>
<tr>
<td></td>
<td>CC DSB</td>
<td>DISABLE</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>UNPROT</td>
<td>UNPROTECT</td>
<td>Reserved for input to the memory protect flip-flop on a given memory board</td>
</tr>
<tr>
<td>21</td>
<td>SS</td>
<td>SINGLE STEP</td>
<td>Used by Front Panel to disable input buffer while panel drives bidirectional data bus</td>
</tr>
<tr>
<td>22</td>
<td>ADDR DSBL</td>
<td>ADDRESS DISABLE</td>
<td>Allows the buffers for the 16 address lines to be tri-stated</td>
</tr>
<tr>
<td>23</td>
<td>DO DSBL</td>
<td>DATA OUT DISABLE</td>
<td>Allows the bidirectional data bus drivers for the 8 data lines to be tri-stated for both input and output data buses</td>
</tr>
<tr>
<td>24</td>
<td>Ø2</td>
<td>Phase 2 Clock</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Ø1</td>
<td>Phase 1 Clock</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PHLDA</td>
<td>Hold Acknowledge</td>
<td>Processor control output signal which appears in response to the HOLD signal;</td>
</tr>
</tbody>
</table>
BUS DEFINITION

Front Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>26(continued)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PWAIT</td>
<td>WAIT</td>
<td>indicates that the data and address bus will go to the high impedance state on the 8080. Note: ADDR DSBL and DO DSBL must be driven to the tri-state the system bus</td>
</tr>
<tr>
<td>28</td>
<td>PINTE</td>
<td>INTERRUPT ENABLE</td>
<td>Processor control output signal which acknowledges that the processor is in a WAIT state</td>
</tr>
<tr>
<td>29</td>
<td>A5</td>
<td>Address Line #5</td>
<td>Processor control output signal indicating interrupts are enabled: may be set or reset by EI and DI instruction and inhibits interrupts from being accepted by the CPU if it is reset</td>
</tr>
<tr>
<td>30</td>
<td>A4</td>
<td>Address Line #4</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>A3</td>
<td>Address Line #3</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>A15</td>
<td>Address Line #15</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>A12</td>
<td>Address Line #12</td>
<td></td>
</tr>
</tbody>
</table>
## BUS DEFINITION

### Front Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>A9</td>
<td>Address Line #9</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>D0</td>
<td>Data Out Line #1</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>D00</td>
<td>Data Out Line #0</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>A10</td>
<td>Address Line #10</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>D04</td>
<td>Data Out Line #4</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>D05</td>
<td>Data Out Line #5</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>D06</td>
<td>Data Out Line #6</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>D12</td>
<td>Data In Line #2</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>D13</td>
<td>Data In Line #3</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>D17</td>
<td>Data In Line #7</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>SM1</td>
<td>M1</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>SOUT</td>
<td>OUT</td>
<td>Status output signal that indicates that the processor is in the fetch cycle for the first byte of an instruction</td>
</tr>
</tbody>
</table>

Status output signal which indicates that the address bus contains the address of an output device and the data bus will contain the output data when PWR is active.
### BUS DEFINITION

#### Front Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>SNIP</td>
<td>INP</td>
<td>Status output signal which indicates that the address bus contains the address of an input device and the input data should be placed on the data bus when PDBIN is active</td>
</tr>
<tr>
<td>47</td>
<td>SMEMR</td>
<td>MEMR</td>
<td>Status output signal which indicates that the data bus will be used for memory read data</td>
</tr>
<tr>
<td>48</td>
<td>SHTLA</td>
<td>HLTA</td>
<td>Status output signal which acknowledges a HALT instruction</td>
</tr>
<tr>
<td>49</td>
<td>CLOCK</td>
<td>CLOCK</td>
<td>2 MHz clock signal</td>
</tr>
<tr>
<td>50</td>
<td>GND</td>
<td>GROUND</td>
<td></td>
</tr>
</tbody>
</table>

#### Back Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>-8V</td>
<td>+8 volts</td>
<td>Unregulated input to 5V regulators</td>
</tr>
<tr>
<td>52</td>
<td>-16V</td>
<td>-16 volts</td>
<td>Negative unregulated voltage</td>
</tr>
<tr>
<td>No.</td>
<td>SYMBOL</td>
<td>NAME</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>53</td>
<td>SSW DSB</td>
<td>SENSE SWITCH DISABLE</td>
<td>Enables the data input buffers so the input from the sense switches may be strobed onto the bidirectional data bus</td>
</tr>
<tr>
<td>54</td>
<td>EXT CLR</td>
<td>EXTERNAL CLEAR</td>
<td>Clear signal for I/O devices (front panel switch closure to ground)</td>
</tr>
<tr>
<td>55</td>
<td>CGND</td>
<td>CHASSIS GROUND</td>
<td></td>
</tr>
<tr>
<td>56-57</td>
<td>UNUSED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>MWRT</td>
<td>MEMORY WRITE</td>
<td>From the Front Panel replacement card indicates that the current data on the Data Out Bus is to be written into the memory location currently on the address bus</td>
</tr>
<tr>
<td>69</td>
<td>PS</td>
<td>PROTECT STATUS</td>
<td>Reserved to indicate the status of the memory protect flip-flop on the memory board currently addressed</td>
</tr>
<tr>
<td>70</td>
<td>PROT</td>
<td>PROTECT</td>
<td>Reserved for input to the memory protect flip-flop on the memory board currently addressed</td>
</tr>
</tbody>
</table>
### BUS DEFINITION

#### Back Side

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>RUN</td>
<td>RUN</td>
<td>Indicates that the RUN/STOP flip-flop is set to run on the front panel.</td>
</tr>
<tr>
<td>72</td>
<td>PRDY</td>
<td>READY</td>
<td>Processor command/control input that controls the run state of the processor; if the line is pulled low the processor will enter a wait state until the line is released.</td>
</tr>
<tr>
<td>73</td>
<td>PINT</td>
<td>INTERRUPT REQUEST</td>
<td>The processor recognizes an interrupt request on this line at the end of the current instruction or while halted. If the processor is in the HOLD state or the Interrupt Enable flip-flop is reset, it will not honor the request.</td>
</tr>
<tr>
<td>74</td>
<td>PHOLD</td>
<td>HOLD</td>
<td>Processor command input signal which requests the processor to enter the HOLD state; allows an external device to gain control of address and data buses as soon as the processor.</td>
</tr>
</tbody>
</table>
### BUS DEFINITION

**Back Side**

<table>
<thead>
<tr>
<th>No.</th>
<th>SYMBOL</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>74</td>
<td></td>
<td>(continued)</td>
<td>has completed its use of these buses for the current machine cycle</td>
</tr>
<tr>
<td>75</td>
<td>PRESET</td>
<td>RESET</td>
<td>Processor command input; while activated the content of the program counter is cleared and the instruction register is set to 0</td>
</tr>
<tr>
<td>76</td>
<td>PSYNC</td>
<td>SYNC</td>
<td>Processor control output provides a signal to indicate the beginning of each machine cycle</td>
</tr>
<tr>
<td>77</td>
<td>PWR</td>
<td>WRITE</td>
<td>Processor control output used for memory write or I/O output control; data on the data bus is stable while the PWR is active</td>
</tr>
<tr>
<td>78</td>
<td>PDBIN</td>
<td>DATA BUS IN</td>
<td>Processor control output signal indicates to external circuits that the data bus is in the input mode</td>
</tr>
<tr>
<td>79</td>
<td>A0</td>
<td>Address Line #0</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>A1</td>
<td>Address Line #1</td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>SYMBOL</td>
<td>NAME</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-----------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>81</td>
<td>A2</td>
<td>Address Line #2</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>A6</td>
<td>Address Line #6</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>A7</td>
<td>Address Line #7</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>A8</td>
<td>Address Line #8</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>A13</td>
<td>Address Line #13</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>A14</td>
<td>Address Line #14</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>A11</td>
<td>Address Line #11</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>D02</td>
<td>Data Out Line #2</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>D03</td>
<td>Data Out Line #3</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>D07</td>
<td>Data Out Line #7</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>D14</td>
<td>Data In Line #4</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>D15</td>
<td>Data In Line #5</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>D16</td>
<td>Data In Line #6</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>D17</td>
<td>Data In Line #1</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>D10</td>
<td>Data In Line #0</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>SINTA</td>
<td>INTA</td>
<td>Status output signal to acknowledge signal for INTERRUPT request</td>
</tr>
<tr>
<td>97</td>
<td>SWO</td>
<td>WO</td>
<td>Status output signal indicates that the operation in the current machine cycle will be a WRITE memory or output function</td>
</tr>
<tr>
<td>No.</td>
<td>SYMBOL</td>
<td>NAME</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>98</td>
<td>SSTACK</td>
<td>STACK</td>
<td>Status output signal indicates that the address bus holds the pushdown stack address from the Stack Pointer</td>
</tr>
<tr>
<td>99</td>
<td>POC</td>
<td>Power-On Clear</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>GND</td>
<td>GROUND</td>
<td></td>
</tr>
</tbody>
</table>
3.2.12 CPU Power Supply - Functional Description

The power supply is designed to be used in the CPU with boards that have on-board regulators. It provides +10 volts and ±18 volts at no load and approximately +7 and ±15.8 volts at full load.

The rectifiers and 115 volt power switching and fusing functions are contained on a small power-supply PC board. Large computer-grade filter capacitors are used to provide minimum ripple. A custom-wound transformer is used to provide the maximum amount of power in the available space to match the exact requirements of the 8080 system. One hundred fifteen volt output terminals are available on the board, switched both fused for such functions as the ventilating fan and auxiliary power outlets on the back panel. A position is provided on the rectifier board for a power switch, for use when the front panel is not installed in the system.

The secondary windings of the power transformer are all connected in series to provide a winding with a center tap and one tap on each side of the center tap midway up the secondary out of which three separate full wave center tap rectifier circuits provide the three required voltages.

Two heavy duty rectifiers are connected to the mid points of the winding and provide +10 volts at the output. A full wave bridge for both plus and negative voltages is connected to the end points of the secondary winding to provide +18 and -18 volts.
Large computer grade electrolytic capacitors are used to filter the rectifier output before applying the voltages to the 8080 back plane.

Since the 7805 regulators used on the 8080 system boards require 7 volts minimum to regulate at 5 volts, a maximum power supply voltage drop of 3 volts is permissible between the no load voltage of 10 volts and the minimum permissible full load voltage of 7 volts.

Many components contribute significantly to the full load power supply voltage drop due to the heavy currents involved. Among these are the resistance of the transformer winding, the resistance of the hook-up wire used in assembling the power supply, the voltage drop across the diode, the voltage drop between cycles from the discharge of the filter capacitor, and contact resistance in any of the joints. A similar situation is true for the ±18 volt supplies; however, because of the lower currents involved, this design situation is less stringent.

Additionally, three series diodes were put in both the +18 and -18 volt power supply leads to hold the no load voltage down to an acceptable value for a number of boards which use a simple resistor and zener diode to regulate for +12 volts.
3.3 **Control Program**

3.3.1 **Program Description**

The control program for the RWLM is written in the PL/M Language (Programming Language/Microcomputers). This language is block structured and somewhat self-documenting. The complete program listing is contained in Appendix C. Most procedures are explained by the comments in the listing.

3.3.2 **The Language**

A PL/M program is a sequence of "declarations" and "executable statements."

The declarations allow the programmer to control allocation of storage, define simple textual substitutions (macros), and define procedures. PL/M is a "block structured" language: procedures may contain further declarations which control storage allocation and define other procedures.

The procedure definition facility of PL/M allows modular programming: a program can be divided into sections (e.g., Teletype input, conversion from binary to decimal forms, and printing output messages). Each of these sections is written as a PL/M procedure.

PL/M handles two kinds of data. Its two basic "data types" are BYTE and ADDRESS. A BYTE variable or constant is one that can be represented as an 8-bit quantity; an ADDRESS variable or constant is a 16-bit or double-byte quantity.
The programmer can DECLARE variable names to represent BYTE or ADDRESS values. One can also declare vectors (or array) of the type of BYTE or ADDRESS.

In general, executable statements specify the computational processes that are to take place. To achieve this, arithmetic, logical (Boolean), and comparison (relational) operators are defined for variables and constants of both types (BYTE and ADDRESS). These operators and operands are combined to form EXPRESSIONS which resemble those of elementary algebra. For example, the PL/M expression

\[ X \times (Y - 3)/R \]

represents this calculation: the value of \( X \) multiplied by the quantity \( Y-3 \), divided by the value of \( R \). Expressions are a major component of PL/M statements. A simple statement which computes a result and stores it in a memory location defined by a variable name. The assignment

\[ Q = X \times (Y - 3)/R \]

first causes the computation to the right of the equal sign, as described above. The result of this computation is then saved in a memory location labeled by the variable name "Q."

Other statements in PL/M perform conditional tests and branching, loop control, and procedure invocation with parameter passing. The flow of program execution is specified by means of control structures that take advantage of the block-structured nature of the language. Input and output statements read and write 8-bit values from and to the 8080 microprocessor's
input and output ports. Procedures can be defined which use these basic input and output statements to perform more complicated I/O operations.

3.3.3 Calibration Coefficients

The Rn-daughter and WL coefficients which were calculated during the calibration runs are stored in the READ ONLY memory of the RWLM in matrix form and are accessed by the CAL procedure on page 9 of the listing. Each constant is allocated 16 bytes of memory starting at location 110 H.

The format of the data is as follows:

ADDRESS (HEX)

| 110-11F | WL  COEF. #1 |
| 120-12F | WL  COEF. #2 |
| 130-13F | WL  COEF. #3 |
| 140-14F | RaA COEF. #1 |
| 150-15F | RaA COEF. #2 |
| 160-16F | RaA COEF. #3 |
| 170-17F | RaB COEF. #1 |
| 180-18F | RaB COEF. #2 |
| 190-19F | RaB COEF. #3 |
| 1A0-1AF | RaC COEF. #1 |
| 1B0-1BF | RaC COEF. #2 |
| 1C0-1CF | RaC COEF. #3 |
Note from the listing that the last character in each data string is a 24H. The math pack interprets this character as the end of a number.

The arrangement of data in this structure allows the routines that calculate the WL and Rn-daughter levels to be tightly coded and it also allows using the same routines to do the calculation for each head with only the change of an indexing variable.

For example, in the CAL procedure the fixed point numbers in the above matrix must be converted to four-byte floating point numbers to be used in the calculations. The code to perform this conversion is as follows:
DO I=0 to 11;
CALL INSTR(.CONSTR + (I*16) + (192 * SELDET),
.FCON + (I*4));
END;

The INSTR routine is passed to the address of the number it is to convert to floating point as the first parameter: CONSTR + (I*16) + (192*SELDET)
when I = 0 and SELDET = 0 (HEAD #1).

The first number converted starts at location 110H, i.e., the first WL coefficient for Head #1. If SELDET = 1 as it would if Head #2 were selected, the first number converted would at location 100H, i.e., 110H+0+(192D x 1) = 1D0H

The INSTR routine then stores the converted number at location pointed by .FCON + (I*4). This, of course, forms a second matrix holding the floating point values of the constant data starting at the address of FCON (21BFH).

The altitude correction factor, the ta counts and the background counts are also converted to floating point and stored by the CAL PROCEDURE.

The calculations are then performed on the data and constants as shown in the listing. The output data is stored in a vector called WL for use by the PRINTREPORT Procedure.

3.4 Floating Point Math Pack

The floating point system consists of a set of subroutines designed to perform arithmetic operations on numeric quantities represented in memory.
The software constituting the floating point system is divided into two sections.

Section 1 contains all the arithmetic routines, while section 2 contains routines which are used to convert data between a binary floating point format and a decimal format suitable for entry or display on input/output equipment.

3.4.1 Floating Point Routines

The 8080 binary floating point system consists of a set of subroutines designed to perform operations on numeric quantities represented in a specific notation. Subroutines are provided to perform a variety of arithmetic and related operations.

The subroutines are designed to be stored and executed in READ-ONLY-MEMORY (ROM) and require 64 bytes of RAM. Scratchpad memory is initialized by a utility subroutine which must be executed before other subroutines are executed for the first time (INT routine called by PL/M).

In general, the subroutines have the following characteristics. Subroutines requiring one operand take it from an internal floating point accumulator. Subroutines requiring two operands take one from the accumulator and the other from the memory location indicated by the contents of the H and L registers upon entry. The numeric result of each operation is stored in the accumulator and is returned to the called in the A, B, C, and D registers.
Upon exit from the arithmetic subroutines, the properties of the result are indicated by the settings of the control bits.

**Carry Bit = 1** The result exceeds the capacity of the accumulator. The other control bits, the contents of the hardware registers, and the contents of the accumulators are meaningless. This situation is also indicated by a non-zero quantity being stored in a flag word.

**Carry Bit = 0** The result is in range. The zero and sign bits are properly set, and the A,B,C, and D registers contain a representation of the value in the accumulator.

**Zero Bit = 1** The result of the operation is zero or a quantity too small to be represented.

**Zero Bit = 0** The result is non-zero.

**Sign Bit = 0** The result is non-zero.

**Sign Bit = 1** The result is negative.

**Sign Bit = 0** The result is positive.

Data are represented in a notation which records eight bits of exponent. One bit of sign, and 24 bits of fraction. The largest magnitude that can be represented is approximately $3.6 \times 10^{38}$. The smallest non-zero magnitude is approximately $2.7 \times 10^{-39}$. The resolution of the notation is approximately $6.2 \times 10^{-8}$, i.e., better than seven-decimal digit precision.
Data values are represented in four consecutive memory words which must be in the same bank of memory. The interpretation of these words is shown below.

Word 1 If non-zero, this word contains the exponent plus a bias of 200 octal. The exponent indicates the power of 2 by which the fraction is multiplied to obtain the represented value. If this word is zero the represented value is zero and words 2, 3, and 4 are meaningless.

Word 2, Bit 7 This bit indicates the sign of the value: 0 if positive, 1 if negative.

Word 2, Bits 6-0 These bits plus as assumed 1 in the bit 7 are the most significant bits of the fraction. The fraction is stored in absolute form (unsigned) with the radix point positioned to the left of bit 7. The value of the fraction is thus less than 1.0 and equal to or greater than 0.5.

Word 3 This word contains the second most significant eight bits of the fraction.

Word 4 This word contains the least significant eight bits of the fraction.

Examples of Data Notation (Octal Notation).

<table>
<thead>
<tr>
<th>Value</th>
<th>Word 1</th>
<th>Word 2</th>
<th>Word 3</th>
<th>Word 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>000</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx x = don't care</td>
</tr>
<tr>
<td>+1.0</td>
<td>201</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>-1.0</td>
<td>201</td>
<td>200</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>+0.1</td>
<td>175</td>
<td>114</td>
<td>314</td>
<td>314</td>
</tr>
<tr>
<td>-100.1</td>
<td>207</td>
<td>310</td>
<td>063</td>
<td>063</td>
</tr>
</tbody>
</table>
3.4.2 Floating Point Accumulator

The floating point accumulator consists of 5 scratchpad words containing, respectively, the accumulator exponent, the accumulator sign, and three words of accumulator fraction. The exponent is recorded with a bias of 200 octal. An exponent word of zero indicates that the value in the accumulator is zero and the remaining words of the accumulator are meaningless. The sign word holds 000 if the accumulator is negative, 200 octal if positive. The fraction is recorded as a normalized positive value with the radix point to the left of the most significant bit of the first fraction word.

3.5 Format Conversion Package

The format conversion package of the 8080 binary floating point system contains subroutines for the conversion of data between the floating point system notation and two other formats. The non-floating point formats are four-word fixed-point format and variable-length character-string format.

The format conversion package is contained in 512 consecutive words of memory (2 banks of ROM) and requires for its execution that the arithmetic and utility packages be available in memory. The combination of this format conversion package and the arithmetic and utility packages uses the first 64 words of a bank of RAM as scratchpad memory.

The fixed point format data processes by this package consist of 32-bit binary numbers occupying four words. (Two's complement notation is used to represent negative values.) The position of the binary point relative to the bits representing the value is denoted by a binary
scaling factor. The binary scaling factor is not normally recorded in
the computer; but, when a format conversion subroutine is called, the
binary scaling factor must be specified (in the E register). A binary
scaling factor of zero indicates the binary point is immediately to the
left of the most significant of the 32 bits representing the value.
A binary scaling factor of 32 indicates the binary point is immediately
to the right of the least significant bit. The permissible range of
the binary scaling factor is -128 (200 octal) to +127 (177 octal).

The character string format data processed by this package consist
of binary representations of characters occupying consecutive words
of memory. A character string may not cross a memory-bank boundary.
The characters which may be included in a character string, and the
corresponding octal representations are listed below.

<table>
<thead>
<tr>
<th>Decimal Digits</th>
<th>008-011B BCD digits*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>360B</td>
</tr>
<tr>
<td>+</td>
<td>373B</td>
</tr>
<tr>
<td>-</td>
<td>375B</td>
</tr>
<tr>
<td>.</td>
<td>376</td>
</tr>
<tr>
<td>Exponential sign</td>
<td>025B</td>
</tr>
</tbody>
</table>

*(These octal representations can be converted to the corresponding
ASCII characters by adding 060B to each.)

The out subroutine generates character strings in two formats. Each
consists of 13 characters. The format used in a specific case is
dependent upon the magnitude of the value represented.

Zero and magnitudes between .1000000 and 9999999. are represented
by a space or minus sign, seven decimal digits and an appropriately
positioned decimal point, and four spaces.
Magnitudes outside the above range are represented by a space or minus sign, a value between 1.000000 and 9.999999, an exponential sign, and a signed two-digit power of ten.

The input subroutine converts character strings in either of the above formats, or a modified version of them. The leading sign character may be included or omitted. Up to 37 digits may be used to indicate the value, with or without an included decimal point. If a power-of-ten multiplier is indicated it may be signed or unsigned and may contain one or two digits. An input character string is terminated by the first character which departs from the specified format.

3.6 Interface Between PL/M and Assembly Language Floating Point Math Pack Routines

There are two sets of routines that interface the PL/M and the assembly language routines. One set of these interfaces appears on page 8 of the PL/M control program listing in Appendix C. The assembly language part of the routine is labeled Math Pack Transfer Routine and is also found in Appendix C.

The PL/M procedures are called by elements in the control program. In turn, they transfer control to the assembly language transfer control to the assembly language transfer routines. These routines move parameters passed by PL/M and make these parameters correspond with the requirements of the Math Pack.
4.0 CALIBRATION

It is evident from (1) in section 2.0 that the numerical values for $N_A$, $N_B$, and $N_C$ depend on $V$, $E_1$, $E_2$, $E_3$, and the corrected counts $T_A$, $T_C$ see (5) and the measured counts $(B + C)$. Therefore, to calibrate the RWLM, $V$, $E_1$, $E_2$, and $E_3$ (the flowrate and the efficiencies of RaA, RaB, and RaC) have to be determined, and one has to ensure that the counts used to calculate $N_A$, $N_B$, and $N_C$ are $T_A$, $(B + C)$, and $T_C$ as defined in section 2.0.

4.1 Measurement of Flowrate $V$

The flowrate, $V$ (liters/min), can easily be measured with a Wet Test Meter (Precision Scientific Co.). The numerical value of $V$ (in liters/min) for the particular detector measured is used in the calibration procedure (see Fig. 8).

4.2 Determinations of $E_1$, $E_2$, and $E_3$

The alpha efficiency of the RWLM detectors, $E_1$, was determined by comparison with a calibrated hemispherical proportional counter. The proportional counter was calibrated with an NBS $^{241}$Am source. An air sample from a plastic drum containing uranium ore was taken. About 40 minutes after the ending of the sampling period the circular portion of the filter strip containing the filtered Rn daughters was cut out. This circular disc was then counted with the proportional counter and with the RWLM. The counting period was always 30 seconds. About 10 measurements with each counter were taken. These results were averaged and $E_1$ was calculated according to the formula:

$$E_1 = \left( \frac{C_{\text{RWLM}}}{C_{\text{PC}}} \right) E_{\text{PC}}$$
where:

\[ C_{\text{RWLM}} = \text{average number of alpha counts observed with RWLM} \]

\[ C_{\text{PC}} = \text{average number of alpha counts observed with the proportional counter.} \]

\[ E_{\text{PC}} = \text{known alpha efficiency of the proportional counter.} \]

The measurements with the two counters should be interlaced. The first measurement with the RWLM, for example, should be followed by the first measurement with the proportional counter, and so on. Counting in this interlaced fashion avoids bias due to decay. The counts from the RWLM detection system can be recorded with an external scaler.

Once \( E_1 \) is known, \( E_2 \) and \( E_3 \) can be determined by the following procedure: Knowledge of \( E_1 \) allows a complete analysis of any air sample by a method using total alpha counts. This method allows calculation of \( N_A, N_B, \) and \( N_C \) for this particular air sample. \( N_A, N_B, \) and \( N_C \) can then be used to calculate the beta disintegrations from RaB and RaC. By comparing the calculated beta disintegrations with the observed total beta counts, the values for \( E_2 \) and \( E_3 \) are obtained. The necessary calculations are all performed by the "RWLM CALIBRATION PROGRAM" (see Fig. 8).

To calculate the beta efficiencies \( E_2 \) and \( E_3 \), the Rn-daughter concentrations (atoms/liter) \( N_A, N_B, \) and \( N_C \) must first be determined from the alpha counts during several counting periods using the following equations:
\[ N_A = 0.962617*\frac{A_s}{E_1}*V \]
\[ N_B = (\frac{-0.919056*A_s - 11.17412*C_s + 2.764731*C_3}{E_1})*V \]  
(2)
\[ N_C = (\frac{0.054135*A_s + 4.305161*C_s - 0.264945*C_3}{E_1})*V \]

where: *

\( A_s = \) RaA counts observed during 5 min starting 13 sec after the end of the 2-min sampling time.

\( C_s = \) RaC' counts observed during the same time interval as above.

\( C_3 = \) RaC' counts observed during 30 min, starting at the same time as above.

The numerical coefficients in (2) are again derived from the laws of radioactive-series decay. This derivation is straightforward but lengthy and will therefore be omitted.

With \( N_A, N_B, \) and \( N_C \) known, \( E_2 \) and \( E_3 \) can be determined from the following equations:

\[ Q_1 = (0.131911*N_A + 0.235122*N_B)*V*E_2 + (0.010901*N_A + 0.029629*N_B + 0.309182*N_C)*V*E_3 \]  
(3)
\[ Q_2 = (0.984198*N_A + 1.046109*N_B)*V*E_2 + (0.389388*N_A + 0.481449*N_B + 1.24961*N_C)*V*E_3 \]

where

\( Q_1 = \) total beta counts observed during 5 min starting 13 sec after the end of the 2-min sampling time.

\( Q_2 = \) total beta counts observed during 30 min same time as above.

*The overlap corrections must be incorporated into the alpha counts (\( A_s, C_s \) and \( C_3 \)). See (5).*
With the beta efficiencies $E_2$ and $E_3$ so determined, (1) can be inverted and properly scaled. When properly scaled this yields a set of equations which gives the Rn-daughter concentrations in pCi/liter. The WL can also be expressed as a linear combination of $A$, $(B + C)$, and $C'$. These counts are net counts, i.e., the background has been subtracted. The equations are of the following form:

$$WL = C_1(A) + C_2(B + C) + C_3(C')$$
$$RaA(pCi/liter) = C_4(A) + C_5(B + C) + C_6(C')$$
$$RaB(pCi/liter) = C_7(A) + C_8(B + C) + C_9(C')$$
$$RaC(pCi/liter) = C_{10}(A) + C_{11}(B + C) + C_{12}(C')$$

$C_1$ through $C_{12}$ are the derived weighting coefficients which are stored in the memory of the RWLM. It is clear from this description of the calibration that $N_A$, $N_B$, and $N_C$ are treated as independent unknowns (i.e., no a priori relationship between these quantities other than radioactive series decay is assumed). The RWLM determines, therefore, the Rn-daughter concentrations and WL without any assumptions about the Rn-daughter equilibrium. Since all weighting coefficients are strictly proportional to the inverse of the flowrate, or $1/V$, a flowrate correction can be made by including a multiplication factor in (4). This implies that a recalibration of the RWLM is unnecessary if it is operated at different altitudes, or any condition which may result in a flowrate different from that used in the calibration, such as a loss of pump efficiency due to wear.
4.3 Discriminator Adjustments

V, E₁, E₂, and E₃ have been determined so far. We will now discuss the adjustments necessary to ensure that the other input parameters VO (overlap), A, (B + C) and C' are properly determined.

VO = fraction of RaC' counts appearing in the RaA channel during the corresponding counting period.

A = total alpha counts in RaA channel during 2 min counting period starting 13 sec after the end of sampling.

(B+C) = total beta counts from RaB and RaC (same period of accumulation as above).

C = total alpha counts in RaC' channel (same as above).

CKUS = total alpha counts from 39 to 41 min used to calculate WLKUS, the Kusnetz WL (with time base starting 13 sec after the end of sampling).

First, the lower alpha discriminator level has to be set above the noise level. This is done easily by gradually increasing the reference voltage level until no background counts due to noise are observed. This adjustment has to be done in an area with negligible airborne radioactivity. In a similar manner the beta channel must be adjusted. A certain number of counts due to ambient gamma radiation will always appear since the beta detector is sensitive to gamma rays. The RWLM channel accumulator registers approximately 250 counts/min for 1 mr/hr of gamma radiation from a $^{226}$Ra source. The next step is the adjustment of the upper alpha discriminator level. For this adjustment,
a multi-channel analyzer (MCA) is needed. The multichannel-analyzer display shows clearly the separation between the RaA and the RaC' alpha peaks. Integrating the combined spectrum between the analyzer's baseline and the channel with the minimum number of counts between the two alpha peaks gives A, the number of counts in the lower alpha channel. Integrating from this channel to the descent of the RaC' spectrum gives C', the number of counts in the upper alpha channel. The upper level alpha discriminator must then be adjusted to obtain agreement between A and C' by comparing the integrated counts under each peak on the MCA with the A and C' counts observed in the alpha channels of the RWLM. This data is collected simultaneously from the same air sample. This agreement can be achieved by gradually adjusting the upper alpha discriminator level in the RWLM and comparing the different sets of A counts and C' counts. An initial minimum setting of the RWLM's upper alpha discriminator can be achieved with a $^{241}$Am standard. This initial setting should be slightly higher than the pulses from the 5.5 MeV $^{241}$alpha's. After the upper alpha discriminator has been properly set, the RWLM will record the quantities A and C' properly. The A counts and C' counts are, however, not yet suitable for the necessary calculations. The A and C' have to be corrected for the degradation of the RaC' spectrum. VO (overlap) is the fraction of the RaC' counts that is detected by the lower alpha channel when a sample is made with the counting delayed 45 min to assure complete decay of RaA. It is algebraically defined by:
VO = (RaC' counts in RaA channel)/(RaC' counts in C channel).

With VO known, all TA and TC counts can be determined from:

\[ TA = A - ((VO) \cdot C') \]
\[ TC' = (1 + VO) \cdot C' \] (5)

These relationships (5) are used in the "RWLM CALIBRATION PROGRAM."

4.4 Calibration Program (see Fig. 8)

The calibration program automates the solution of the equations for radioactive-series decay, as applied to the RWLM, and calculates the required efficiencies and coefficients. This program is available in both the FORTRAN and BASIC languages. The program performs the following functions:

1) It calculates the Rn-daughter concentrations by three different methods and the WL by four different methods from the same air sample.

   The four different methods are:
   a) The alpha-spectroscopic method
   b) The total-alpha method
   c) The Kusnetz method
   d) The IWLM/RWLM method

2) It calculates the RaB and RaC efficiencies using the Rn-daughter concentrations calculated by the total-alpha method, and the beta counts at two different periods of time.

3) It calculates the 12 coefficients needed for the IWLM/RWLM method. See (4).

4) It calculates the Rn-daughter concentrations from 9 of these calculated coefficients, in units of atoms/liter, for
comparison to the other two methods, and also in units of pCi/liter which is the unit used by the RWLM.

5) It calculates the WL from the Rn-daughter concentrations, and with the use of the three remaining coefficients it also determines the WL directly from the measured counts.

In order to understand the program, it will be necessary to refer to the calibration program and to the "Table of Definitions of Symbols in the calibration program" (see Appendix C and Fig. 8). The program has two branches, one which calculates beta efficiencies from the input data, and another which uses the mean value of the beta efficiencies to derive the final weighted coefficients (see Figs. 9 and 10). In both branches the Rn-daughter concentrations are calculated by using the alpha-spectroscopic method (line 140) and the total-alpha method (line 150).

Lines 50 through 52 are the coefficients for these two equations. Lines 100 through 120 are statements that adapt the input data for these methods. Line 170 calculates the WL with the alpha-spectroscopic method. Line 180 uses the total-alpha method for the same calculation. Line 190 calculates the WL by the Kusnetz method. These three WL calculations are available for comparison with the IWLM/RWLM method to be explained later.

As shown in Fig. 9, the first question asked by the computer is "Calculate or Input $E_B$, $E_C$ ($E_1$, $E_2$)." When the operator types a "C", the program is routed into Branch 1 (lines 2000 to 2100)
which uses the inverted equations calculate the beta efficiencies for RaB and RaC (E_2 and E_3, respectively) by using the observed beta counts (Q_1 and Q_2) and the Rn-daughter concentrations previously determined by the total-alpha method. If an "I" is entered (see Fig. 10), the input branch is selected and the beta efficiencies are entered as data (line 210).

Statements 3000 to 3210 perform a matrix inversion and give the final weighting coefficients C_1 through C_{12} for (4) after conversion to the proper units.

The Rn-daughter concentrations are calculated in units of atoms/liter by the statements for FO(1), FO(2), and FO(3) and are printed out to allow comparison with the values calculated by the alpha methods. FO(1), FO(2), and FO(3) are then converted to units of pCi/liter (line 260) and printed. The WL is calculated from the Rn-daughter concentrations which were derived using the IWLM/RWLM method (line 270) and directly using the coefficients of the IWLM/RWLM method (line 280). Note that these two different equations give exactly the same results. See Fig. 9, "WL from IWLM (direct and from Rn daughters)."

4.5 Calibration Procedure

The calibration procedure consists of the following steps: first, 10 sets of data are automatically collected; second, the RWLM calibration program is used to calculate the beta efficiencies; third, the RWLM calibration program is used to calculate the coefficients for the RWLM using the mean value of the calculated beta efficiencies; and, fourth, these new coefficients
TODAY IS 00/00/00 THE TIME IS 00:00
DATE CORRECT (Y/N)? N
ENTER YEAR-77 ENTER MONTH-03 ENTER DAY-21 ENTER HOUR-11 ENTER MINUTE-45

TODAY IS 03/21/77 THE TIME IS 11:45
DO YOU WANT DETECTOR 1 RUN (Y/N)? Y
DO YOU WANT DETECTOR 2 RUN (Y/N)? Y

ENTER TIME INTERVAL BETWEEN SAMPLES HOURS-00 MIN-20
ENTER TIME TO START HOUR-12 MIN-00
INPUT TIME TO STOP
ENTER YEAR-77 ENTER MONTH-03 ENTER DAY-22 ENTER HOUR-00 ENTER MINUTE-00

Figure 7 Data Terminal Dialog between Operator and System
1 REM RWLM CALIBRATION PROGRAM
2 REM AEGONNE NATIONAL LABORATORY ELECTRONICS DIVISION
20 PRINT "RWLMCAL.RAS VAR:1": GOSUR.4000
30 FOR I=1 TO 8
35 READ C(I),CO(I)
40 NEXT I
50 DATA .926838,1.918637,0,-3.196779,0,2.506266
60 DATA -.879403,1.701921,-11.12606,-11.62053,2.752640,16.756924
70 DATA A5=2.623113,23.825945,0.579584,16.756924
80 Ti=45+V0*C5:T2=(1+V0)*C5
90 JS=1:5=S1=S2+S3
100 T5=V0+V+(1+V0)*C5
110 FOR 0=0 TO 6 STEP 3
120 FL(0)=C5(0)+C5(0)+T3/11
130 NEXT 0
140 W(S)=C5(0)+1/11
150 IF Y=1 THEN GOSUR 2000: GOTO 200
200 INPUT "Env E C PLEASE":E2,E3
300 PRINT "R L M CAL I R A T I O N P R O G R A M"
400 PRINT "FNA="F1(0),"FNI="F1(1),"FNC="F1(2)
500 PRINT "W L 2="W2
600 PRINT "R N D A U G H T E R S" "F N ( R W L M ) I N A T O M S"
700 PRINT F0(1),F0(2),F0(3)
800 PRINT "F N D A U G H T E R S" "F N ( R W L M ) I N P C I / L"
900 PRINT P1,P2,P3
100 PRINT "W L F R O M I W L ( D I R E C T A N D F R O M R N - D A U G H T E R S)"
110 PRINT P4
120 PRINT "W L C O E F F I C I E N T S F O R I W L - M E M O R Y"'
130 PRINT "A 1(1)=.5R03H6*E1*V:A2(1)=.036204*E2V+.1584*E3*V
140 A2(2)=.098134*E2*V+.026941*E3*V:A3(1)=.001584*EI*V
150 RETURN
300 A1(1)=.5R03H6*E1*V:A2(1)=.036204*E2V+.1584*E3*V
160 A2(2)=.098134*E2*V+.026941*E3*V:A3(1)=.001584*EI*V
170 Figure 8 RWLM Calibration Program
Figure 8. (Continued)
Figure 9 RWLM Calibration Program - Readout with calculate (C) beta efficiency branch
RWLMCAL.BAS V02.1
CALCULATE OR INPUT ER, EC (C/I)? 1
DATE, PLACE? 8/4/76, QUIRK 1
FLOW RATE (LITERS/MIN)? 11.2
EFFICIENCY OF ALPHA DETECTOR? .20
OVLAP? .18
TOTAL BETAS FROM RAB AND RAC DURING 5 MINUTES? 56416.5
TOTAL BETAS FROM RAB AND RAC DURING 30 MINUTES? 294583
TOTAL ALPHAS FROM RAA CHANNEL DURING 30 MINUTES? 5419
TOTAL ALPHAS FROM RAC' CHANNEL DURING 35 MINUTES? 107085
TOTAL ALPHAS FROM RAC' CHANNEL DURING 5 MINUTES? 10151.5
TOTAL ALPHAS FROM RAC' CHANNEL DURING 5 MINUTES? 16980
TOTAL ALPHAS FROM RAC' CHANNEL DURING 30 MINUTES? 94182
TOTAL BETAS FROM RAB AND RAC DURING 5 MINUTES? 28898
TOTAL ALPHAS FROM RAC' CHANNEL DURING 2 MINUTES? 5127
TOTAL ALPHAS FROM RAC' CHANNEL DURING 2 MINUTES? 6998
TOTAL ALPHAS FROM RAC' CHANNEL DURING 2 MINUTES? 29185
ER, EC PLEASE? .205, .3865

RWLM CALIBRATION

DATE PLACE
8/4/76 QUIRK 1

FNA = 2935.72  FNB = 34272.6  FNC = 25533.5

FNA2 = 3029.67  FNB2 = 32472.7  FNC2 = 25368.5

WL1 = 3.84209  WLKUS = 3.99554  ER = .205  EC = .3865

WL2 = 3.73619

RN-DAUGHTER COEFFICIENTS FOR IWL-MEMORY
-0.0787421  0  -0.0141736
-3.30608E-03  .0517072  -117316
9.48931E-05  -3.72704E-03  .072216

RN-DAUGHTER (RWLM) IN ATOMS
2998.51  29746.1  26508.4

RN-DAUGHTER (RWLM) IN PCI/L
306.957  346.555  420.134

WL FROM IWL (DIRECT AND FROM RN-DAUGHTERS)
3.65888  3.65888

WL COEFFICIENTS FOR IWL-MEMORY
6.48517E-05  2.4A304E-04  -3.40269E-04

NORMAL STOP AT LINE 500
BREAK IN 500 OK

Figure 10 RWLM Calibration Program – Readout with input (I)
beta efficiency branch
### Sample Calibration Data Readout

**Today is 08/04/76**  
The time is 16:45  
Altitude correction factor = 1

<table>
<thead>
<tr>
<th>Working Level</th>
<th>3.66579</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radium A</td>
<td>306.957</td>
</tr>
<tr>
<td>Radium B</td>
<td>352.161</td>
</tr>
<tr>
<td>Radium C</td>
<td>419.73</td>
</tr>
</tbody>
</table>

**Sample Counts** | **Background Counts** | **Net Counts**
---|---|---
Radium A | 5184 | 27 | 5157.0
Radium (B+C) | 23569 | 671 | 22898.0
Radium C | 7015 | 22 | 6993.0

**5 Minute Count**

<table>
<thead>
<tr>
<th>Sample Counts</th>
<th>Background Counts</th>
<th>Net Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radium A</td>
<td>10219</td>
<td>10151.5</td>
</tr>
<tr>
<td>Radium (B+C)</td>
<td>50594</td>
<td>5641.5</td>
</tr>
<tr>
<td>Radium C</td>
<td>17035</td>
<td>16990.0</td>
</tr>
</tbody>
</table>

**30 Minute Count**

<table>
<thead>
<tr>
<th>Sample Counts</th>
<th>Background Counts</th>
<th>Net Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radium A</td>
<td>85884</td>
<td>85419.0</td>
</tr>
<tr>
<td>Radium (B+C)</td>
<td>304648</td>
<td>32943.0</td>
</tr>
<tr>
<td>Radium C</td>
<td>94512</td>
<td>94182.0</td>
</tr>
</tbody>
</table>

**35 Minute Count**

<table>
<thead>
<tr>
<th>Sample Counts</th>
<th>Background Counts</th>
<th>Net Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radium A</td>
<td>27678</td>
<td>27405.5</td>
</tr>
<tr>
<td>Radium (B+C)</td>
<td>344188</td>
<td>332445.5</td>
</tr>
<tr>
<td>Radium C</td>
<td>107400</td>
<td>107045.0</td>
</tr>
</tbody>
</table>

**39-41 Minute Count**

<table>
<thead>
<tr>
<th>Sample Counts</th>
<th>Background Counts</th>
<th>Net Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radium A</td>
<td>762</td>
<td>735.0</td>
</tr>
<tr>
<td>Radium (B+C)</td>
<td>13850</td>
<td>13179.0</td>
</tr>
<tr>
<td>Radium C</td>
<td>4657</td>
<td>4635.0</td>
</tr>
</tbody>
</table>

Figure 11 Sample Calibration Data Readout
are loaded into the RWLM memory. This procedure assumes that the alpha efficiency, the flowrate, and the RaC' overlap were previously determined. The first step is initiated by entering a "C" command via the keyboard. The RWLM then proceeds automatically to complete 10 successive calibration sampling runs (see Fig. 11).

When the calibration mode is selected by typing a "C" command, the following events occur:

1) a normal sampling run is taken, but the accumulators are not reset at the end of the 2-min counting time.*
2) the total count for 5 min is recorded.
3) the total count for 30 min is recorded.
4) the total count for 35 min is recorded.
5) the accumulators are reset and the total count from 39 to 41 min is recorded.

This procedure is automatically repeated a total of 10 times for each head in order to obtain a statistically sufficient number of runs. It is possible to acquire the data for a complete calibration run in approximately 14 hours. A calibration run can be completed overnight. Unlike the earlier IWLM which had no microprocessor, the RWLM does not require an operator to be present to record the data. A sample calibration run is shown in Fig. 11.

On completion of the data collection, the RWLM calibration program is run, using the collected data to calculate beta efficiencies $E_i$.

*The zero time-base is selected as 13 sec after the end of the sampling period.
and $E_2$. The beta efficiencies calculated by the program are averaged to obtain the mean beta efficiencies.

The program is then rerun in the input branch (see Fig. 10) to calculate the final coefficients $C_1$ through $C_{12}$ for the RWLM equations.

These calculated coefficients $C_1$ through $C_{12}$ are then programmed into the RWLM PROM memory and the calibration is complete. Note that since the RWLM has two detection heads, two individual calibrations are required for the total system.
5.0 TEST RESULTS

The measurements of WL and Rn-daughter concentrations taken with the RWLM in the Twilight Mine on November 16-18, 1976, are tabulated in Table 5.1. The agreement of the WL, N_A, N_B, and N_C values obtained during the calibration runs is very good. WL is compared with three different and independent methods of measurement. One sample is used to obtain data for all the different methods of measurement. The WL's as measured by the total-alpha method, alpha-spectroscopic method, and the RWLM method show consistently good agreement. Occasionally the WL as measured by Kusnetz method reads a higher value than the other three methods; however, the WL using the Kusnetz method is in good agreement with the other three methods in more than 80% of the measurements. The Rn-daughter concentrations as measured by the RWLM show an absence of bias which can be seen easily by inspection of the tabulated data in Table 5.1.
<table>
<thead>
<tr>
<th>Method</th>
<th>WL</th>
<th>RaA</th>
<th>RaB</th>
<th>RaC</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha spectroscopic</td>
<td>1.640</td>
<td>2187</td>
<td>15496</td>
<td>8365</td>
</tr>
<tr>
<td>total alpha</td>
<td>1.610</td>
<td>2112</td>
<td>14799</td>
<td>8687</td>
</tr>
<tr>
<td>Kusnetz</td>
<td>1.686</td>
<td>2163</td>
<td>15330</td>
<td>8664</td>
</tr>
<tr>
<td>RWLM</td>
<td>1.623</td>
<td>2116</td>
<td>14168</td>
<td>8569</td>
</tr>
<tr>
<td>Det 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run #1 Time:</td>
<td>1353</td>
<td>1.686</td>
<td>2163</td>
<td>15330</td>
</tr>
<tr>
<td>Run #2 Time:</td>
<td>1439</td>
<td>1.645</td>
<td>2163</td>
<td>15330</td>
</tr>
<tr>
<td>Run #3 Time:</td>
<td>1525</td>
<td>1.607</td>
<td>2216</td>
<td>15110</td>
</tr>
<tr>
<td>Run #4 Time:</td>
<td>1630</td>
<td>1.656</td>
<td>2112</td>
<td>14799</td>
</tr>
<tr>
<td>Run #5 Time:</td>
<td>1716</td>
<td>1.640</td>
<td>2216</td>
<td>15110</td>
</tr>
<tr>
<td>Run #6 Time:</td>
<td>1801</td>
<td>1.574</td>
<td>2153</td>
<td>14595</td>
</tr>
<tr>
<td>Run #7 Time:</td>
<td>1847</td>
<td>1.566</td>
<td>2327</td>
<td>15615</td>
</tr>
<tr>
<td>Run #8 Time:</td>
<td>1933</td>
<td>1.557</td>
<td>2095</td>
<td>15159</td>
</tr>
<tr>
<td>Run #9 Time:</td>
<td>2019</td>
<td>1.481</td>
<td>2157</td>
<td>13351</td>
</tr>
</tbody>
</table>

*Table 5.1: Data Calculated from Calibration Runs in Twilight Mine*
<table>
<thead>
<tr>
<th>Date</th>
<th>Det</th>
<th>Run</th>
<th>Time</th>
<th>WL</th>
<th>RaA</th>
<th>RaB</th>
<th>RaC</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/16/76</td>
<td>Det 1</td>
<td>Run #10</td>
<td>2105</td>
<td>1.470</td>
<td>2037</td>
<td>13760</td>
<td>7488</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.472</td>
<td>1998</td>
<td>13731</td>
<td>7624</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.593</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.460</td>
<td>2051</td>
<td>13471</td>
<td>7597</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 1</td>
<td>Run #11</td>
<td>2150</td>
<td>1.437</td>
<td>2026</td>
<td>13280</td>
<td>7434</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.454</td>
<td>1999</td>
<td>13553</td>
<td>7498</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.537</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.474</td>
<td>2040</td>
<td>13916</td>
<td>7404</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 1</td>
<td>Run #12</td>
<td>2236</td>
<td>1.459</td>
<td>2075</td>
<td>13380</td>
<td>7628</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.459</td>
<td>1971</td>
<td>13190</td>
<td>8002</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.598</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.510</td>
<td>2101</td>
<td>14447</td>
<td>7379</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 1</td>
<td>Run #13</td>
<td>2322</td>
<td>1.427</td>
<td>2118</td>
<td>13266</td>
<td>7120</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.428</td>
<td>2040</td>
<td>13137</td>
<td>7396</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.484</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.463</td>
<td>2121</td>
<td>13854</td>
<td>7130</td>
<td>RWLM</td>
</tr>
<tr>
<td>11/17/76</td>
<td>Det 2</td>
<td>Run #1</td>
<td>0008</td>
<td>1.761</td>
<td>2762</td>
<td>16266</td>
<td>8628</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.752</td>
<td>2743</td>
<td>16056</td>
<td>8710</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.322</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.759</td>
<td>2752</td>
<td>16587</td>
<td>8288</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 2</td>
<td>Run #2</td>
<td>0053</td>
<td>1.810</td>
<td>2800</td>
<td>17298</td>
<td>8359</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.761</td>
<td>2515</td>
<td>15874</td>
<td>9462</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.832</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.801</td>
<td>2801</td>
<td>17053</td>
<td>8436</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 2</td>
<td>Run #3</td>
<td>0139</td>
<td>1.793</td>
<td>2805</td>
<td>16725</td>
<td>8633</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.800</td>
<td>2768</td>
<td>16786</td>
<td>8751</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.878</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.827</td>
<td>2804</td>
<td>17378</td>
<td>8552</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 2</td>
<td>Run #4</td>
<td>0225</td>
<td>1.790</td>
<td>2881</td>
<td>16404</td>
<td>8764</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.813</td>
<td>2867</td>
<td>16812</td>
<td>8772</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.813</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.776</td>
<td>2849</td>
<td>16779</td>
<td>8810</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 2</td>
<td>Run #5</td>
<td>0311</td>
<td>1.788</td>
<td>2827</td>
<td>15973</td>
<td>9250</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.838</td>
<td>2933</td>
<td>17104</td>
<td>8781</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.890</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.864</td>
<td>2831</td>
<td>17246</td>
<td>9263</td>
<td>RWLM</td>
</tr>
<tr>
<td></td>
<td>Det 2</td>
<td>Run #6</td>
<td>0356</td>
<td>1.852</td>
<td>2903</td>
<td>17142</td>
<td>9042</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.875</td>
<td>2902</td>
<td>17565</td>
<td>9005</td>
<td>total alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.574</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.841</td>
<td>2897</td>
<td>16721</td>
<td>9278</td>
<td>RWLM</td>
</tr>
<tr>
<td>Date</td>
<td>Det 2 Run #7</td>
<td>Time</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>Method</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>11/17/76</td>
<td>1.858</td>
<td>0442</td>
<td>2903</td>
<td>16689</td>
<td>9593</td>
<td>Alpha spectroscopic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.884</td>
<td></td>
<td>2905</td>
<td>17181</td>
<td>9537</td>
<td>Total alpha</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.932</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.886</td>
<td></td>
<td>2950</td>
<td>16745</td>
<td>9922</td>
<td>RWLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Det 2 Run #8</th>
<th>Time: 0528</th>
<th>WL</th>
<th>RaA</th>
<th>RaB</th>
<th>RaC</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.844</td>
<td>2858</td>
<td>17030</td>
<td>9097</td>
<td>Alpha spectroscopic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.877</td>
<td>2911</td>
<td>17744</td>
<td>8849</td>
<td>Total alpha</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.017</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.886</td>
<td>2852</td>
<td>18003</td>
<td>8843</td>
<td>RWLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Det 2 Run #9</th>
<th>Time: 0614</th>
<th>WL</th>
<th>RaA</th>
<th>RaB</th>
<th>RaC</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.888</td>
<td>3053</td>
<td>16553</td>
<td>9967</td>
<td>Alpha spectroscopic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.957</td>
<td>3187</td>
<td>18084</td>
<td>9365</td>
<td>Total alpha</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.062</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.953</td>
<td>3032</td>
<td>17769</td>
<td>9886</td>
<td>RWLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Det 2 Run #10</th>
<th>Time: 0700</th>
<th>WL</th>
<th>RaA</th>
<th>RaB</th>
<th>RaC</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.916</td>
<td>2894</td>
<td>17874</td>
<td>9409</td>
<td>Alpha spectroscopic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.880</td>
<td>2674</td>
<td>16793</td>
<td>10259</td>
<td>Total alpha</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.994</td>
<td></td>
<td></td>
<td></td>
<td>Kusnetz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.913</td>
<td>2875</td>
<td>17450</td>
<td>9818</td>
<td>RWLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Data Calculated from Calibration Runs in Twilight Mine

<table>
<thead>
<tr>
<th>Date: 11/17/76</th>
<th>Det 1</th>
<th></th>
<th></th>
<th></th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run #1</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1529</td>
<td>0.333</td>
<td>640</td>
<td>3055</td>
<td>1438</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1529</td>
<td>0.311</td>
<td>562</td>
<td>2510</td>
<td>1756</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1529</td>
<td>0.342</td>
<td>657</td>
<td>2882</td>
<td>1514</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1529</td>
<td>0.329</td>
<td>657</td>
<td>2882</td>
<td>1514</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #2</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1615</td>
<td>0.330</td>
<td>650</td>
<td>3027</td>
<td>1402</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1615</td>
<td>0.323</td>
<td>648</td>
<td>2895</td>
<td>1422</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1615</td>
<td>0.325</td>
<td>669</td>
<td>2752</td>
<td>1447</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1615</td>
<td>0.318</td>
<td>669</td>
<td>2752</td>
<td>1447</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #3</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1701</td>
<td>0.306</td>
<td>596</td>
<td>2766</td>
<td>1348</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1701</td>
<td>0.299</td>
<td>587</td>
<td>2633</td>
<td>1388</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1701</td>
<td>0.312</td>
<td>611</td>
<td>2519</td>
<td>1405</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1701</td>
<td>0.296</td>
<td>611</td>
<td>2519</td>
<td>1405</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #4</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1746</td>
<td>0.293</td>
<td>554</td>
<td>2772</td>
<td>1201</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1746</td>
<td>0.265</td>
<td>466</td>
<td>2083</td>
<td>1567</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1746</td>
<td>0.374</td>
<td>527</td>
<td>2629</td>
<td>1188</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1746</td>
<td>0.281</td>
<td>527</td>
<td>2629</td>
<td>1188</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #5</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1832</td>
<td>0.290</td>
<td>532</td>
<td>2727</td>
<td>1237</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1832</td>
<td>0.279</td>
<td>512</td>
<td>2483</td>
<td>1327</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1832</td>
<td>0.269</td>
<td>513</td>
<td>2499</td>
<td>1310</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1832</td>
<td>0.279</td>
<td>513</td>
<td>2499</td>
<td>1310</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #6</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 1918</td>
<td>0.267</td>
<td>513</td>
<td>2265</td>
<td>1343</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1918</td>
<td>0.276</td>
<td>557</td>
<td>2509</td>
<td>1173</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 1918</td>
<td>0.278</td>
<td>505</td>
<td>2036</td>
<td>1406</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 1918</td>
<td>0.257</td>
<td>505</td>
<td>2036</td>
<td>1406</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #7</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 2004</td>
<td>0.253</td>
<td>492</td>
<td>2257</td>
<td>1147</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2004</td>
<td>0.250</td>
<td>469</td>
<td>2163</td>
<td>1235</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2004</td>
<td>0.267</td>
<td>500</td>
<td>2250</td>
<td>1123</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 2004</td>
<td>0.252</td>
<td>500</td>
<td>2250</td>
<td>1123</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #8</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 2050</td>
<td>0.255</td>
<td>475</td>
<td>2456</td>
<td>1017</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2050</td>
<td>0.245</td>
<td>441</td>
<td>2211</td>
<td>1155</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2050</td>
<td>0.244</td>
<td>477</td>
<td>2545</td>
<td>933</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 2050</td>
<td>0.256</td>
<td>477</td>
<td>2545</td>
<td>933</td>
<td>RWLM</td>
</tr>
<tr>
<td>Run #9</td>
<td>WL</td>
<td>RaA</td>
<td>RaB</td>
<td>RaC</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td>Time: 2135</td>
<td>0.222</td>
<td>469</td>
<td>1894</td>
<td>1021</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2135</td>
<td>0.221</td>
<td>464</td>
<td>1864</td>
<td>1042</td>
<td>total alpha</td>
</tr>
<tr>
<td>Time: 2135</td>
<td>0.228</td>
<td>464</td>
<td>1597</td>
<td>1074</td>
<td>RWLM</td>
</tr>
<tr>
<td>Time: 2135</td>
<td>0.207</td>
<td>464</td>
<td>1597</td>
<td>1074</td>
<td>RWLM</td>
</tr>
<tr>
<td>Date</td>
<td>Det 1</td>
<td>Det 2</td>
<td>Det 2</td>
<td>Det 2</td>
<td>Det 2</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>11/17/76</td>
<td>Run #10</td>
<td>Run #1</td>
<td>Run #2</td>
<td>Run #3</td>
<td>Run #4</td>
</tr>
<tr>
<td>Time:</td>
<td>2221</td>
<td>2301</td>
<td>2353</td>
<td>0038</td>
<td>0124</td>
</tr>
<tr>
<td>WL</td>
<td>0.213</td>
<td>1.90</td>
<td>1.85</td>
<td>1.864</td>
<td>1.893</td>
</tr>
<tr>
<td>RaA</td>
<td>431</td>
<td>2871</td>
<td>2847</td>
<td>2937</td>
<td>2848</td>
</tr>
<tr>
<td>RaB</td>
<td>1889</td>
<td>17707</td>
<td>16555</td>
<td>16752</td>
<td>17795</td>
</tr>
<tr>
<td>RaC</td>
<td>942</td>
<td>9268</td>
<td>9690</td>
<td>9569</td>
<td>9167</td>
</tr>
<tr>
<td>Method</td>
<td>alpha spectroscopic</td>
<td>total alpha</td>
<td>alpha spectroscopic</td>
<td>total alpha</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td>Kusnetz</td>
<td>RWLM</td>
<td>Kusnetz</td>
<td>RWLM</td>
<td>Kusnetz</td>
</tr>
<tr>
<td>11/18/76</td>
<td>Run #9</td>
<td>Run #10</td>
<td>Run #10</td>
<td>Run #10</td>
<td>Run #10</td>
</tr>
<tr>
<td>Time:</td>
<td>0513</td>
<td>0559</td>
<td>0559</td>
<td>0559</td>
<td>0559</td>
</tr>
<tr>
<td>WL</td>
<td>1.983</td>
<td>1.985</td>
<td>1.985</td>
<td>1.985</td>
<td>1.985</td>
</tr>
<tr>
<td>RaA</td>
<td>3022</td>
<td>2976</td>
<td>2976</td>
<td>2976</td>
<td>2976</td>
</tr>
<tr>
<td>RaB</td>
<td>18304</td>
<td>18546</td>
<td>18546</td>
<td>18546</td>
<td>18546</td>
</tr>
<tr>
<td>RaC</td>
<td>9874</td>
<td>9750</td>
<td>9750</td>
<td>9750</td>
<td>9750</td>
</tr>
<tr>
<td>Method</td>
<td>alpha spectroscopic</td>
<td>total alpha</td>
<td>alpha spectroscopic</td>
<td>total alpha</td>
<td>alpha spectroscopic</td>
</tr>
<tr>
<td></td>
<td>Kusnetz</td>
<td>RWLM</td>
<td>Kusnetz</td>
<td>RWLM</td>
<td>Kusnetz</td>
</tr>
</tbody>
</table>
6.0 **OPERATING INSTRUCTIONS**

6.1 **Normal Operation**

6.1.1 Load the filter-paper transport mechanism. This is accomplished by removing the alpha-detector hold-down screw and swinging the detector assembly up. The feed and takeup reel covers can then be removed and the reels freed from the instrument by pulling the release plunger for each reel. Note that the reels are identical. The filter paper is then loaded onto the feed reel in such a manner that it is pulled from the top of the reel as it feeds to the sampling port (see Fig. 6). The paper is then placed in the guide slots, under the paper-advance bar, and attached to the takeup reel with a small piece of tape. The takeup reel is then advanced manually several revolutions to assume proper action. If the paper advance bar is not in the highest position, type a "J" on the terminal.

6.1.2 Assure that both the signal cable and the power cable are properly connected to the detector assemblies, and that these cables are also connected to the appropriate channel connector on the CPU.

6.1.3 Apply power to the central processing unit and the TI Silent 700 terminal.

6.1.4 Assure that the AC power is ON in the remote unit by observing the red lamp above the name plate.

6.1.5 If the lamp does not come on in the power is applied, the internal switch is in the "off" position. Remove the cover plate and place the switch in the "on" position.
6.1.6 Place the control switch on the terminal to the "ON-LINE" position and press the restart button. The terminal should respond with:
REMOTE WORKING LEVEL MONITOR VERSION 11.0
TODAY IS XX/XX/XX THE TIME IS XX:XX

6.1.7 When first energized, the contents of the data memory locations are unspecified; to reset the system type:
R
The system will respond:
REMOTE WORKING LEVEL MONITOR VERSION 11.0
TODAY IS 00/00/00 THE TIME IS 00:00

6.1.8 To set the date and time, type
D
The system will respond:
ENTER YEAR-XX ENTER MONTH-XX ENTER DAY-XX
ENTER HOUR-XX ENTER MINUTE-XX
The operator must always enter a two-digit number, i.e., January is 01, etc. If a mistake is made in entry, continue typing or press the restart button and re-enter the "D": command. When all data has been entered, the system will respond:
TODAY IS XX/XX/XX THE TIME IS XX:XX
Note that the time function assumes a 24-hour clock.
When in the command loop (indicated by an asterisk on the terminal), an "F" can be types to check the date and time.
6.1.9 If an altitude-correction factor other than 1.0 is desired, the operator can now type an "A". The system will respond:

SELECT HEAD 1 OR 2 -

The operator then types 1 or 2.

The system responds:

ENTER ALTITUDE CORRECTION FACTOR=

The operator then types in an appropriate number such as 1.053 or 0.9562. The sequence repeats to allow the operator to set the correction factor on the remaining head and then returns to the command loop.

6.1.10 The system is now ready for normal operation. To enter this mode the operator types a "G".

The system responds:

TODAY IS XX/XX/XX THE TIME IS XX:XX DATE CORRECT (Y/N)?

If date is correct, type "Y"; if not, type "N" and system will enter correction routine. When date is correct, system responds: DO YOU WANT DETECTOR 1 RUN (Y/N)? and waits for operator response.

After response the system types -

DO YOU WANT DETECTOR 2 RUN (Y/N)?

The system responds: (Operator response is indicated as Y or N)

ENTER TIME INTERVAL BETWEEN SAMPLES HOURS-XX MINS-XX

(Time interval must be greater than 14 minutes.)
The system now enters the sampling loop and remains in it until:

a) the time interval is complete.
b) an error is detected.
c) the reset button on the terminal is depressed.

Fig. 7 shows a complete dialogue between an operator and the system.

6.2 Source Check Mode

6.2.1 In order to provide a first-order check of the system functions, a source check mode has been provided. To use this mode the alpha-detector hold-down screw must be removed and the source holder inserted between the detectors. When it is properly inserted, it cannot be removed without slightly lifting the alpha detector holder assembly.

6.2.2 When the source holder is in place in the desired head or heads, type "s" to enter the Source Check Mode.

The system will respond with:

SELECT HEAD 1 OR 2 - 1
SELECT MODE A - B - C - A
The system now proceeds to accumulate a two-minute count in the detector RaA channel. At the end of the counting time it will print a report. All normal calculations will be performed with the count data in the unselected channels set to 0.

This mode allows the operator to assure that the system is operating normally as the calculations for all source check modes should be consistent within statistical variances from day to day.

6.3 Calibration Mode

The calibration mode which has been described in section 4.0 is entered by typing a "C".
7.0 REFERENCES

An Instant Working Level Meter with Automatic Individual Radon Daughter Readout for Uranium Mines,
P. G. Groer, D. J. Keefe, W. P. McDowell, and R. F. Selman,

Rapid Determination of Radon Daughter Concentrations and Working Level with the Instant Working Level Meter
P. G. Groer, D. J. Keefe, W. P. McDowell, and R. F. Selman,
Int. Symp. on Radiation Protection in Mining and Milling of Uranium and Thorium, Bordeaux, France, 1974 (September).

A Portable Alpha Counter for Uranium Mines with Preset, Updated Readout,
D. J. Keefe, W. P. McDowell, and P. G. Groer,
Int. Symp. on Radiation Protection in Mining and Milling of Uranium and Thorium, Bordeaux, France, 1974 (September).

Interfacing Calculator Chips to Nuclear-Counting Systems,
W. P. McDowell, E. J. Keefe, and P. G. Groer,
APPENDIX A

This appendix contains the drawings of all mechanical parts. Copies of these drawings can be obtained from the reproducible master.

TABLE OF CONTENTS
FILTER TRANSPORT ASSEMBLY
MODEL 1A

<table>
<thead>
<tr>
<th>Part #</th>
<th>Drawing #</th>
<th>Part Name and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G0230-0003-PL-00</td>
<td>Parts List</td>
</tr>
<tr>
<td>2</td>
<td>G0230-0003-DE-00</td>
<td>Assembly Drawing</td>
</tr>
<tr>
<td>3</td>
<td>G0230-0004-DC-00</td>
<td>Top Guide Plate</td>
</tr>
<tr>
<td>4</td>
<td>G0240-0005-DC-00</td>
<td>Right Side Plate</td>
</tr>
<tr>
<td>5</td>
<td>G0230-0006-DC-00</td>
<td>Left Side Plate</td>
</tr>
<tr>
<td>6</td>
<td>G0230-0007-DC-00</td>
<td>Actuating Plate Housing</td>
</tr>
<tr>
<td>7</td>
<td>G0230-0008-DC-00</td>
<td>Actuating Plate</td>
</tr>
<tr>
<td>8</td>
<td>G0230-0009-DC-00</td>
<td>Spool</td>
</tr>
<tr>
<td>9</td>
<td>G0230-0010-DC-00</td>
<td>Gear Housing</td>
</tr>
<tr>
<td>10</td>
<td>G0230-0011-DB-00</td>
<td>Gear Housing Shaft</td>
</tr>
<tr>
<td>11</td>
<td>G0230-0012-DB-00</td>
<td>Actuating Plate Cam</td>
</tr>
<tr>
<td>12</td>
<td>G0230-0013-DB-00</td>
<td>Actuating Plate Drive Rack</td>
</tr>
<tr>
<td>13</td>
<td>G0230-0014-DB-00</td>
<td>Actuating Plate Retaining Bar</td>
</tr>
<tr>
<td>14</td>
<td>G0230-0015-DB-00</td>
<td>Spool Plunger Spring</td>
</tr>
<tr>
<td>15</td>
<td>G0230-0016-DB-00</td>
<td>Plunger Knob</td>
</tr>
<tr>
<td>16</td>
<td>G0230-0017-DB-00</td>
<td>Spool Retaining Plunger</td>
</tr>
<tr>
<td>17</td>
<td>G0230-0018-DB-00</td>
<td>Retaining Plunger Housing</td>
</tr>
<tr>
<td>18</td>
<td>G0230-0019-DB-00</td>
<td>Spacer Washer</td>
</tr>
<tr>
<td>19</td>
<td>G0230-0020-DB-00</td>
<td>Spool Shaft</td>
</tr>
<tr>
<td>20</td>
<td>G0230-0021-DB-00</td>
<td>Spool Bearing Housing</td>
</tr>
<tr>
<td>21</td>
<td>G0230-0022-DB-00</td>
<td>Spacer</td>
</tr>
<tr>
<td>22</td>
<td>G0230-0023-DB-00</td>
<td>Idler Gear Shaft</td>
</tr>
<tr>
<td>23</td>
<td>G0230-0024-DB-00</td>
<td>Drive Gear Bearing Housing</td>
</tr>
<tr>
<td>24</td>
<td>G0230-0025-DB-00</td>
<td>Drive Gear Shaft</td>
</tr>
<tr>
<td>25</td>
<td>G0230-0026-DB-00</td>
<td>Lower Tie Bar</td>
</tr>
<tr>
<td>26</td>
<td>G0230-0027-DB-00</td>
<td>Roller</td>
</tr>
<tr>
<td>27</td>
<td>G0230-0028-DB-00</td>
<td>Roller Shaft</td>
</tr>
<tr>
<td>28</td>
<td>G0230-0029-DB-00</td>
<td>Actuating Plate Roller Shaft</td>
</tr>
<tr>
<td>29</td>
<td>G0230-0030-DB-00</td>
<td>Hinge Pin</td>
</tr>
<tr>
<td>30</td>
<td>G0230-0031-DB-00</td>
<td>Mounting Plate Hinge Piece</td>
</tr>
<tr>
<td>31</td>
<td>G0230-0032-DB-00</td>
<td>Mounting Plate Hinge Bracket</td>
</tr>
<tr>
<td>32</td>
<td>G0230-0033-DB-00</td>
<td>Mounting Plate</td>
</tr>
<tr>
<td>33</td>
<td>G0230-0034-DB-00</td>
<td>Pressure Plate</td>
</tr>
<tr>
<td>34</td>
<td>G0230-0035-DB-00</td>
<td>Pressure Plate Guide Screw</td>
</tr>
<tr>
<td>35</td>
<td>G0230-0036-DB-00</td>
<td>Pressure Plate Support Spring</td>
</tr>
<tr>
<td>36</td>
<td>G0230-0037-DB-00</td>
<td>Stand off</td>
</tr>
<tr>
<td>37</td>
<td>G0230-0038-DB-00</td>
<td>Stand Off Shield</td>
</tr>
<tr>
<td>38</td>
<td>G0230-0039-DB-00</td>
<td>Stand Off Shield Insert</td>
</tr>
<tr>
<td>39</td>
<td>G0230-0040-DB-00</td>
<td>Shielding Cover</td>
</tr>
<tr>
<td>40</td>
<td>G0230-0041-DB-00</td>
<td>Cover Upper Shield Slug</td>
</tr>
<tr>
<td>41</td>
<td>G0230-0042-DB-00</td>
<td>Cover Lower Shield Disc</td>
</tr>
<tr>
<td>42</td>
<td>G0230-0043-DB-00</td>
<td>Tube Holder</td>
</tr>
<tr>
<td>43</td>
<td>G0230-0044-DB-00</td>
<td>Tube Holder Insert</td>
</tr>
<tr>
<td>44</td>
<td>G0230-0045-DB-00</td>
<td>Tube</td>
</tr>
<tr>
<td>Part #</td>
<td>Drawing #</td>
<td>Part Name and Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>43</td>
<td>G0230-0046-DB-00</td>
<td>Tube End Cap</td>
</tr>
<tr>
<td>44</td>
<td>G0230-0047-DB-00</td>
<td>Spacer Ring</td>
</tr>
<tr>
<td>45</td>
<td>G0230-0048-DB-00</td>
<td>Coil Bobbin</td>
</tr>
<tr>
<td>46</td>
<td>G0230-0049-DB-00</td>
<td>Supply Spool Shaft</td>
</tr>
<tr>
<td>47</td>
<td>G0230-0050-DB-00</td>
<td>Friction Washer</td>
</tr>
<tr>
<td>48</td>
<td>G0230-0051-DB-00</td>
<td>Tape Tension Adjustment Spring</td>
</tr>
<tr>
<td>49</td>
<td>G0230-0052-DB-00</td>
<td>Source Holder</td>
</tr>
<tr>
<td>50</td>
<td>G0230-0053-DB-00</td>
<td>Coil Shield Washer</td>
</tr>
<tr>
<td>51</td>
<td>G0230-0054-DB-00</td>
<td>Led Proximity Sensor Collet</td>
</tr>
<tr>
<td>52</td>
<td>G0230-0055-DB-00</td>
<td>Retaining Cap</td>
</tr>
<tr>
<td>53</td>
<td>G0230-0056-DB-00</td>
<td>Led Proximity sensor collet Housing</td>
</tr>
<tr>
<td>54</td>
<td>G0230-0057-DB-00</td>
<td>Counter Shield</td>
</tr>
<tr>
<td>55</td>
<td>G0230-0058-DB-00</td>
<td>Skanner Spacer</td>
</tr>
<tr>
<td>56</td>
<td>G0230-0059-DB-00</td>
<td>Worm 32P. 4th'Ds, 438PD</td>
</tr>
</tbody>
</table>
**APPENDIX B**

**TABLE OF CONTENTS**

ANL - ELECTRONICS DIVISION  
DRAWING # EL-G-7171

**REMOTE WORKING LEVEL MONITOR (RWLM)**

<table>
<thead>
<tr>
<th>Sheet #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RWLM Head Control &amp; Select Boards (S100 Buss Compatible)</td>
</tr>
<tr>
<td>2</td>
<td>RWLM Accumulators (S100 Buss Compatible)</td>
</tr>
<tr>
<td>3</td>
<td>RWLM Microprocessor Chassis (Conn. Interconnections)</td>
</tr>
<tr>
<td>4</td>
<td>Priority Interrupt</td>
</tr>
<tr>
<td>5</td>
<td>Real Time Clock</td>
</tr>
<tr>
<td>6</td>
<td>Tachometer Voltage - vs - Flowrate</td>
</tr>
<tr>
<td>7</td>
<td>RWLM Interconnection Wiring Diagram (Detector Chassis)</td>
</tr>
<tr>
<td>8</td>
<td>Alpha Channel Voltage Amplifier</td>
</tr>
<tr>
<td>9</td>
<td>RWLM SLO-SYN Translator</td>
</tr>
<tr>
<td>10</td>
<td>RWLM Beta - Preamplifier</td>
</tr>
<tr>
<td>11</td>
<td>RWLM Photomultiplier - Bias Network</td>
</tr>
<tr>
<td>12</td>
<td>RWLM Remote Detector Power Supplies</td>
</tr>
<tr>
<td>13</td>
<td>RWLM Misc. Cont. Card (5PC 787)</td>
</tr>
<tr>
<td>14</td>
<td>RWLM Line-Drivers &amp; Isolation Line Receivers</td>
</tr>
<tr>
<td>15</td>
<td>RWLM Misc. Cont. Card (5PC788)</td>
</tr>
<tr>
<td>16</td>
<td>RWLM Discriminators &amp; Single Channel Analyzer</td>
</tr>
</tbody>
</table>

*This appendix contains the electronic circuit diagrams.  
Copies of these diagrams can be obtained from the reproducible master.*
APPENDIX C

This appendix contains the computer programs used in the development and calibration of the RWLM.

1. RWLM Floating Point Math Pack - Arithmetic and Utility Routines
2. RWLM Floating Point Math Pack - Format Conversion Routines
3. Math Pack Transfer Routine
4. Remote Working Level Monitor Control Program
5. RWLM Calibration Program
6. Table of Definitions of Symbols in the Calibration Program
7. Rn-daughter and Working Level Coefficient Program
PUM FLOATING POINT MATH PACK-ARITHMETIC AND UTILITY ROUTINES

LINE ADDRESS B1 B2 B3 B4 ERROR SOURCE INTEL 8080 ASSEMBLER PAGE 1
00000036 SYSTIN
00000037 SYSTIN
00000038 SYSTIN
00000039 SYSTIN
00000040 SYSTIN
00000041 SYSTIN
00000042 SYSTIN
00000043 SYSTIN
00000044 SYSTIN
00000045 SYSTIN
00000046 SYSTIN
00000047 SYSTIN
00000048 SYSTIN
00000049 SYSTIN
00000050 SYSTIN
00000051 SYSTIN
00000052 SYSTIN
00000053 SYSTIN
00000054 SYSTIN
00000055 SYSTIN
00000056 SYSTIN
00000057 SYSTIN
00000058 SYSTIN

1 3200
2 3200 60 32
3 3200 32 00
4 3200 22 00
5 3200 CO 22

ARITH EQU 32H ; BANK NUMBER OF ARITH PKG
ARNTH EQU $32H
SCP EQU 2200H
SCRF EQU 22H ; BANK NUMBER OF SCRATCH

; 8080 BINARY FLOATING POINT SYSTEM
; PROGRAMMER CAL OP
; DATE 26 DECEMBER 1973
; ARITH IS THE BEGINNING ADDRESS OF THE
; ARITHMETIC AND UTILITY PACKAGE OF THE FLOATING
; POINT SYSTEM.
; SCP IS THE BEGINNING ADDRESS OF THE
; RAM USED AS SCP AT CPD FOR THE SYSTEM.
; THE RAM MULTIPLY AND DIVIDE SUBROUTINES
; ARE MOVED FROM B0 TO RAM BY SUBROUTINE
; INIT AND ARE EXECUTED IN RAM ONLY.
; RAM MULTIPLY SUBROUTINE.
; RAM DIVIDE SUBROUTINE.

MULX EQU $ARITH+SCR
MULP3 ADI 0 ; ADD OPERAND 3RD FRACTION
MULP2 ADI 0 ; ADD OPERAND 2ND FRACTION
MULP1 ADI 0 ; ADD OPERAND 1ST FRACTION
MULX5 ADI 0 ; ADD DIVISOR 3RD FRACTION
MULX4 ADI 0 ; ADD DIVISOR 2ND FRACTION
MULX3 ADI 0 ; ADD DIVISOR 1ST FRACTION
MULX2 ADI 0 ; ADD OPERAND 3RD FRACTION
MULX1 ADI 0 ; ADD OPERAND 2ND FRACTION
MULX0 ADI 0 ; ADD OPERAND 1ST FRACTION
DIVX5 EQU $ARITH+SCR
DIVX4 EQU $ARITH+SCR
DIVX3 EQU $ARITH+SCR
DIVX2 EQU $ARITH+SCR
DIVX1 EQU $ARITH+SCR
DIVX0 EQU $ARITH+SCR

OP4S EQU $ARITH+SCR
OP3S EQU $ARITH+SCR
OP2S EQU $ARITH+SCR
OP1S EQU $ARITH+SCR
OP0S EQU $ARITH+SCR

DIVX6 EQU $ARITH+SCR
DIVX5 EQU $ARITH+SCR
DIVX4 EQU $ARITH+SCR
DIVX3 EQU $ARITH+SCR
DIVX2 EQU $ARITH+SCR
DIVX1 EQU $ARITH+SCR
DIVX0 EQU $ARITH+SCR

ADD ADD OPERAND 3RD FRACTION
SUB SUBDIVISION 4TH FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
ADD ADD DIVISOR 3RD FRACTION
SUB SUBDIVISION 1ST FRACTION
MUL MULTIPLY 1ST FRACTION
DIV DIVIDE 1ST FRACTION
ADD ADD OPERAND 2ND FRACTION
SUB SUBDIVISION 2ND FRACTION
MUL MULTIPLY 2ND FRACTION
DIV DIVIDE 2ND FRACTION
ADD ADD OPERAND 1ST FRACTION
SUB SUBDIVISION 3RD FRACTION
MUL MULTIPLY 3RD FRACTION
DIV DIVIDE 3RD FRACTION
PACK-ARITHMETIC AND UTILITY ROUTINES

CH3: SF EQU ACC3+1 ; SUBTRACTION FLAG

INT1: MOV E,M

INIT: MOV P,E

SF EQU ACC3+1 ; SUBTRACTION FLAG

NOV A,E

INIT: MOV P,E

RET: RETURN TO CALLER

STORE SUBROUTINE ENTRY POINT.

ZR0: MOV H,SCR ; TO ADDRESS SCRATCH BANK

ZRT: MOV H,ACCE ; TO ADDR ACCUM EXPON

XRA A ; ZERO

FLOATING POINT ZRO SUBROUTINE EMT. PNT.

FLOATING POINT CHS SUBROUTINE EMT. PNT.

FLOATING POINT ABS SUBROUTINE EMT. PNT.
BIMS FLOATING POINT MATH PACK-ARITHMETIC AND UTILITY ROUTINES

<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1 B2 B3 B4</th>
<th>ERROR</th>
<th>SOURCE</th>
<th>INTEL 6800 ASSEMBLER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>3250</td>
<td>AF</td>
<td></td>
<td></td>
<td>Abs: XRA A :</td>
<td>ZERO</td>
</tr>
<tr>
<td>106</td>
<td>3251</td>
<td>26 22</td>
<td></td>
<td></td>
<td>MVI R, SCR :</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>3252</td>
<td>2F 31</td>
<td></td>
<td></td>
<td>MVI L, ACCS :</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>3255</td>
<td>A5</td>
<td></td>
<td></td>
<td>ANA H :</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>3256</td>
<td>2E 80</td>
<td></td>
<td></td>
<td>FCOMQ :</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>3258</td>
<td>77</td>
<td></td>
<td></td>
<td>MOV Q, A :</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV R, SCPF :</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANA A :</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JZ 0 :</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MVI D, A :</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV L, M :</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV L, M :</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
</tbody>
</table>

BIMS FLOATING POINT MATH PACK-ARITHMETIC AND UTILITY ROUTINES

<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1 B2 B3 B4</th>
<th>ERROR</th>
<th>SOURCE</th>
<th>INTEL 6800 ASSEMBLER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>3250</td>
<td>AF</td>
<td></td>
<td></td>
<td>Abs: XRA A :</td>
<td>ZERO</td>
</tr>
<tr>
<td>106</td>
<td>3251</td>
<td>26 22</td>
<td></td>
<td></td>
<td>MVI R, SCR :</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>3252</td>
<td>2F 31</td>
<td></td>
<td></td>
<td>MVI L, ACCS :</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>3255</td>
<td>A5</td>
<td></td>
<td></td>
<td>ANA H :</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>3256</td>
<td>2E 80</td>
<td></td>
<td></td>
<td>FCOMQ :</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>3258</td>
<td>77</td>
<td></td>
<td></td>
<td>MOV Q, A :</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV R, SCPF :</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A, M :</td>
<td></td>
</tr>
</tbody>
</table>
BULK FLOATING POINT MATH PACK: ARITHMETIC AND UTILITY ROUTINES

<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>ERROR</th>
<th>SOURCE</th>
<th>INTEL 8080 ASSEMBLER PAGE 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>209</td>
<td>32E0</td>
<td>26</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAVE INITIAL EXPONENT.</td>
</tr>
<tr>
<td>210</td>
<td>32E2</td>
<td>2E</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI H,SCRB : TO ADDRESS SCRATCH BANK</td>
</tr>
<tr>
<td>211</td>
<td>32E5</td>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI L,ACCE : TO ADDP ACCUM EXPONENT</td>
</tr>
<tr>
<td>212</td>
<td>32E6</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A.M : ACCUMULATOR EXPONENT</td>
</tr>
<tr>
<td>213</td>
<td>32E7</td>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DCR L : TO ADDR INITIAL EXPONENT</td>
</tr>
<tr>
<td>214</td>
<td>32E8</td>
<td>67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV H.A : INITIAL EXPONENT</td>
</tr>
<tr>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CHECK FOR ZERO OPERAND.</td>
</tr>
<tr>
<td>216</td>
<td>32E9</td>
<td>7B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A,F : OPERAND EXPONENT</td>
</tr>
<tr>
<td>217</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANA A : SET CONTROL BITS</td>
</tr>
<tr>
<td>218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JZ TST1 : IF OPERAND IS ZERO</td>
</tr>
<tr>
<td>219</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JST GENERATE SUBTRACTION FLAG, RESTORE</td>
</tr>
<tr>
<td>220</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUPPRESSED FRACTION BIT.</td>
</tr>
<tr>
<td>221</td>
<td>32EC</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV L.B : OPERAND SIGN AND 1ST FRACTN</td>
</tr>
<tr>
<td>222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A.B : OPERAND SIGN AND 1ST FRACTION</td>
</tr>
<tr>
<td>223</td>
<td>32ED</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ORB 200Q : OPERAND 1ST FRACTION</td>
</tr>
<tr>
<td>224</td>
<td>32EF</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV B.A : OPERAND 1ST FRACTION</td>
</tr>
<tr>
<td>225</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XRA L : OPERAND SIGN</td>
</tr>
<tr>
<td>226</td>
<td>32F0</td>
<td>AD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI L,ACCS : TO ADDRESS ACCUMULATOR SIGN</td>
</tr>
<tr>
<td>227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XRA M : SUBTRACTION FLAG</td>
</tr>
<tr>
<td>228</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI L,SP : TO ADDRESS SUBTRACTION FLAG</td>
</tr>
<tr>
<td>229</td>
<td>32F2</td>
<td>2E</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV H.A : SUBTRACTION FLAG</td>
</tr>
<tr>
<td>230</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DETERMINING RELATIVE MAGNITUDES OF</td>
</tr>
<tr>
<td>231</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OPERAND AND ACCUMULATOR</td>
</tr>
<tr>
<td>232</td>
<td>32F3</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV L,ACCE : TO ADDRESS ACCUMULATOR EXPONENT</td>
</tr>
<tr>
<td>233</td>
<td>32F4</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A.M : ACCUMULATOR EXPONENT</td>
</tr>
<tr>
<td>234</td>
<td>32F5</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANA A : SET CONTROL BITS</td>
</tr>
<tr>
<td>235</td>
<td>32F6</td>
<td>CA</td>
<td>86</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>JZ ADD17 : IF ACCUMULATOR IS ZERO</td>
</tr>
<tr>
<td>236</td>
<td>32F7</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUB E : DIFFERENCE IN EXPONENTS</td>
</tr>
<tr>
<td>237</td>
<td>3300</td>
<td>DA</td>
<td>0E</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>JC ADD2 : IF ACCUM SMALLER THAN OP</td>
</tr>
<tr>
<td>238</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CHECK FOR INsignificant OFFRAND.</td>
</tr>
<tr>
<td>239</td>
<td>3303</td>
<td>FA</td>
<td>5B</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td>JN TST1 : IF THE OPERAND IS INsignificant</td>
</tr>
<tr>
<td>240</td>
<td>3306</td>
<td>FE</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPI 031Q : COMPARE SHIFT COUNT TO 25</td>
</tr>
<tr>
<td>241</td>
<td>330C</td>
<td>DA</td>
<td>2D</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>JC ADD17 : OPERAND IS INsignificant</td>
</tr>
<tr>
<td>242</td>
<td>330C</td>
<td>C3</td>
<td>5B</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td>JNP TST1 : OPERAND IS INsignificant</td>
</tr>
<tr>
<td>243</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ADD2 : IF ACCUM IS INsignificant</td>
</tr>
<tr>
<td>244</td>
<td>33EC</td>
<td>#2</td>
<td>86</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>ADX2 : JP ADD17 : IF ACCUM IS INsignificant</td>
</tr>
<tr>
<td>245</td>
<td>3311</td>
<td>PE</td>
<td>E7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPI 347Q : COMPARE SHIFT COUNT TO 3455 25</td>
</tr>
<tr>
<td>246</td>
<td>3313</td>
<td>DA</td>
<td>86</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>JC ADD17 : IF ACCUM IS INsignificant</td>
</tr>
<tr>
<td>247</td>
<td>3316</td>
<td>73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV M.E : OPERAND EXPONENT</td>
</tr>
<tr>
<td>248</td>
<td>3317</td>
<td>5F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV E.A : SHIFT COUNT</td>
</tr>
<tr>
<td>249</td>
<td>3318</td>
<td>2E</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI L.SP : TO ADDRESS THE SUBTRACTION FLAG</td>
</tr>
<tr>
<td>250</td>
<td>331A</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A.E : SUBTRACTION FLAG</td>
</tr>
<tr>
<td>251</td>
<td>331B</td>
<td>2E</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI L,ACCS : TO ADDRESS THE ACCUMULATOR SIGN</td>
</tr>
<tr>
<td>252</td>
<td>331D</td>
<td>AE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XRA N : OPERAND SIGN</td>
</tr>
<tr>
<td>253</td>
<td>331E</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NVI H.A : ACCUMULATOR SIGN</td>
</tr>
<tr>
<td>254</td>
<td>331F</td>
<td>AF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XRA A : ZEROFRO</td>
</tr>
<tr>
<td>255</td>
<td>3320</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUBB E : COMPLEMENT SHIFT COUNT</td>
</tr>
<tr>
<td>256</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCHANGE ACCUMULATOR AND OPERAND.</td>
</tr>
<tr>
<td>257</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV M.E : ACCUMULATOR 1ST FRACTION</td>
</tr>
<tr>
<td>258</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV E.A : OPERAND 1ST FRACTION</td>
</tr>
<tr>
<td>259</td>
<td>3323</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV B.E : ACCUMULATOR 1ST FRACTION</td>
</tr>
</tbody>
</table>
BFMI FLOATING POINT MATH PACK—ARITHMETIC AND UTILITY Routines
BUILT FLOATING POINT MATH PACK—ARITHMETIC AND UTILITY ROUTINES

COO00336 STSIN
COO00337 STSIN
COO00338 STSIN
COO00339 STSIN
COO00340 STSIN
COO00341 STSIN
COO00342 STSIN
COO00343 STSIN
COO00344 STSIN
COO00345 STSIN
COO00346 STSIN
COO00347 STSIN
COO00348 STSIN
COO00349 STSIN
COO00350 STSIN
COO00351 STSIN
COO00352 STSIN
COO00353 STSIN
COO00354 STSIN
COO00355 STSIN
COO00356 STSIN
COO00357 STSIN
COO00358 STSIN
COO00359 STSIN
COO00360 STSIN
COO00361 STSIN
COO00362 STSIN
COO00363 STSIN
COO00364 STSIN
COO00365 STSIN
COO00366 STSIN
COO00367 STSIN
COO00368 STSIN
COO00369 STSIN
COO00370 STSIN
COO00371 STSIN
COO00372 STSIN
COO00373 STSIN
COO00374 STSIN
COO00375 STSIN
COO00376 STSIN
COO00377 STSIN
COO00378 STSIN
COO00379 STSIN
COO00380 STSIN
COO00381 STSIN
COO00382 STSIN
COO00383 STSIN
COO00384 STSIN
COO00385 STSIN
COO00386 STSIN
COO00387 STSIN
COO00388 STSIN

LINE ADDR B1 B2 B3 B4 ERROR SOURCE INTEL 8080 ASSEMBLY PAGE 7

313 3363 2D DCR L ; TO ADDRESS MINUEND 2ND FRACTION
314 3364 7E MOV A,H ; MINUEND 2ND FRACTION
315 3365 99 SUB A ; SUBTRAEND 2ND FRACTION
316 3366 4F MOV C,A ; DIFFERENCE 2ND FRACTION
317 3367 2D DCR L ; TO ADDRESS MINUEND 1ST FRACTION
318 3368 7E MOV A,H ; MINUEND 1ST FRACTION
319 3369 98 SUB B ; SUBTRAEND 1ST FRACTION
320 336A 67 MOV B,A ; DIFFERENCE 1ST FRACTION
321 336B 0C EP 33 ADD10: CP COMP ; COMPLEMENT IF NEGATIVE
322 336C F4 02 34 CP NORM ; NORMALIZE IF NECESSARY
323 336D F2 48 32 JP ZR01 ; IF UNDERFLOW OR ZERO
324 336E CD 30 34 ADD11: CALL ROND ; CALL ROUNDBOUND SUBROUTI
325 336F DA CA 32 JC OVERF ; IF OVERFLOW
326 3370 97 SUB M ; DIFFERENCE IN EXPONENTS
327 3371 2D 2F MOV L,A ; DIFFERENCE IN EXPONENTS
328 3372 7B MOV A,E ; ACCUMULATOREXponent
329 3373 96 MOV B,A ; ACCUM SIGN AND 1ST FPC
330 3374 6F MOV L,A ; DIFFERENCE IN EXPONENTS
331 3375 7B MOV A,E ; ACCUMULATOREXponent
332 3376 01 SUB M ; DIFFERENCE IN EXPONENTS
333 3377 3C MOV A,E ; ACCUMULATOREXponent
334 3378 4D MOV C,L ; SIGNIFICANCE INDEX
335 3379 C9 RET ; RETURN TO CALLER
336 337A 35 MOV L,F ; TO ADDRESS PREV EXPONENT
337 337B 2E 35 MOV A,M ; SUBTRACTION FLAG
338 337C 3E MOV L,F ; TO ADDRESS ACCUMULATOR SIGN
339 337D DC 32 MOV A,M ; SUBTRACTION FLAG
340 337E 02 DCR L ; TO ADDR ACCUM EXponent SIGN
341 337F 2D MOV A,M ; SUBTRACTION FLAG
342 3380 CD 3C 32 CALL STHO ; SET OPERAND SIGN
343 3381 98 MOV A,M ; SUBTRACTION FLAG
344 3382 C3 7A 33 IRR L,ACCS ; ADD ACCUMULATOR SIGN
345 3383 00 JMP ADD12 ; JOIN EXIT CODE
346 3384 3C ADD17: MOV L,F ; TO ADDR SUBTRACTION FL
347 3385 A7 MOV A,M ; EXPONENT MODIFIER
348 3386 2C MOV L,F ; TO ADDR OF SIGN, 1ST FPC
349 3387 4E MOV C,M ; OPERAND SIGN AM 1ST FRACTION
350 3388 02 MOV L,F ; TO ADDRESS OPERAND 2ND FRACTION
351 3389 56 MOV C,M ; OPERAND SIGN AM 2ND FRACTION
352 338A 2C MOV L,F ; TO ADDRESS OPERAND 3RD FRACTION
353 338B 4E MOV C,M ; OPERAND SIGN AM 3RD FRACTION
354 338C 26 22 MOV L,F ; TO ADDRESS OPERAND 4TH FRACTION
355 338D 2E 3C MOV L,F ; TO ADDRESS OPERAND 5TH FRACTION
356 338E 7E MOV L,F ; TO ADDRESS ACCUMULATOR EXponent
357 338F 98 MOV A,M ; ACCUMULATOREXponent
358 3390 A7 MOV A,M ; ACCUMULATOREXponent
359 3391 C6 MOV A,M ; ACCUMULATOREXponent
360 3392 00 ADD B ; RESULT EXPONENT PLUS BIAS
361 3393 47 MOV B,A ; RESULT EXPONENT PLUS BIAS
362 3394 78 MOV A,B ; RESULT EXPONENT PLUS BIAS
FLOATING POINT ARITHMETIC AND UTILITY ROUTINES

**Line ADDR B1 B2 B3 B4 ERROR SOURC**  
**E INTEL 8080 ASSEMBLER PAGE 8**

```
0000391 SYSIN  365 33A8 06 80  MOV E,\200Q ; EXP BIAS, SIGN MASK, MS BIT
0000392 SYSIN  366 33AA P2 B8 33  JF OVUN ; IF OVERFLOW OR UNDERFLOW
0000393 SYSIN  367 33A0 09 80  SUB E, 90 ; REMOVE EXCESS EXP BIAS
0000394 SYSIN  368 33AE C8  MOV RC ; RETURN IF UNDERFLOW
0000395 SYSIN  369 33AF 77  MOV H, A ; RESULT EXPONENT
0000396 SYSIN  370 33B0 2C  INR L ; TO ADDRESS ACCUMULATOR SIGN
0000397 SYSIN  371 33B1 7E  MOV H, A ; ACCUMULATOR SIGN
0000398 SYSIN  372 33B2 A9  MOV XRA C ; RESULT SIGN IN SIGN BIT
0000399 SYSIN  373 33B3 A0  MOV A ; RESULT SIGN
0000400 SYSIN  374 33B4 77  MOV H, A ; RESULT SIGN
0000401 SYSIN  375 33B5 79  MOV A, C ; OPERAND SIGN AND 1ST FRACT
0000402 SYSIN  376 33B6 80  SUB E, 1 ; OPERAND 1ST FRACTION
0000403 SYSIN  377 33B7 C9  RET ; RETURN TO CALLER
0000404 SYSIN  378 33B8 07  OUT ; SUBROUTINE TO LEFT SHIFT THE B, C,
0000405 SYSIN  379 33B9 08  BD ; D, AND E REGISTERS ONE BIT.
0000406 SYSIN  380 33BA AF  LSH : MOV A, E ; ORIGINAL CONTENTS OF F
0000407 SYSIN  381 33BB C9  LSH1: MOV A, D ; ORIGINAL CONTENTS OF D
0000408 SYSIN  382 33BC 7B  RAL ; LEFT SHIFT P
0000409 SYSIN  383 33BD 17  MOV E, A ; RESTORE CONTENTS OF E REGISTER
0000410 SYSIN  384 33BE 5F  LSH : MOV A, E ; ORIGINAL CONTENTS OF F
0000411 SYSIN  385 33BF 7A  LSH1: MOV A, D ; ORIGINAL CONTENTS OF D
0000412 SYSIN  386 33C0 17  RAL ; LEFT SHIFT D
0000413 SYSIN  387 33C1 77  MOV D, A ; RESTORE CONTENTS OF D REGISTER
0000414 SYSIN  388 33C2 75  MOV B, C ; ORIGINAL CONTENTS OF C REGISTER
0000415 SYSIN  389 33C3 17  RAL ; LEFT SHIFT C
0000416 SYSIN  390 33C4 4F  MOV C, A ; RESTORE CONTENTS OF C REGISTER
0000417 SYSIN  391 33C5 78  MOV A, B ; ORIGINAL CONTENTS OF B REGISTER
0000418 SYSIN  392 33C6 8F  ADC A ; LEFT SHIFT B
0000419 SYSIN  393 33C7 C7  MOV B, A ; RESTORE CONTENTS OF B REGISTER
0000420 SYSIN  394 33C8 C9  RET ; RETURN TO CALLER
0000421 SYSIN  395 33C9 1E 00  RIGHT SHIFT THE B, C, D AND E REGISTERS
0000422 SYSIN  396 33CA 7B  TO THE SHIFT COUNT IN THE A REGISTER
0000423 SYSIN  397 33CB 2E 08  ; SHIFT OPERAND TO REGISTER INDICATED BY
0000424 SYSIN  398 33CD 8D  ; SHIFT COUNT
0000425 SYSIN  399 33CE FA DA 33  RSH : MOV E, 0 ; OPERAND 4TH FRACT IS ZERO
0000426 SYSIN  400 33CF 00  RSH0 : MOV L, C10Q ; EACH REG IS 8 BITS OF
0000427 SYSIN  401 33D0 18  PSH1 : CMP L ; COMPARE SHIFT COUNT TO 8
0000428 SYSIN  402 33D1 2A  JM RSH2 ; IF REQ SHIFT LESS THAN 8
0000429 SYSIN  403 33D2 45  MOV E, D ; OPERAND 4TH FRACTION
0000430 SYSIN  404 33D3 6A  MOV P, C ; OPERAND 3RD FRACTION
0000431 SYSIN  405 33D4 48  MOV C, 3 ; OPERAND 2ND FRACTION
0000432 SYSIN  406 33D4 69  MOV B, 0 ; OPERAND 1ST FRACTION IS ZERO
0000433 SYSIN  407 33D5 95  SUB L ; REDUCE SHIFT COUNT BY 1 REG
0000434 SYSIN  408 33D6 0B  JM RSH1 ; IF MORE SHIFTS REQUIRED
0000435 SYSIN  409 33D7 C2 33  ; SHIFT OPERAND RIGHT BY -SHIFT COUNT-
0000436 SYSIN  410 33D8 C3  ; BITS.
0000437 SYSIN  411 33D9 A7  RSH2 : ANA A ; SET CONTROL BITS
0000438 SYSIN  412 33DA 47  RZ ; RETURN IF SHIFT COMPLETE
0000439 SYSIN  413 33DB C8  MOV L, A ; SHIFT COUNT
0000440 SYSIN  414 33DC 6F  RSH3 : ANA A ; CLEAR CARRY BIT
```

BULK FLOATING POINT MAC PACK-ARITHMETIC AND UTILITY ROUTINES

<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>ERROR</th>
<th>SOURCE</th>
<th>INTEL 8080</th>
<th>ASSEMBLED</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000496</td>
<td>STIS</td>
<td>417</td>
<td>33DF</td>
<td>7F</td>
<td>00000497</td>
<td>STIS</td>
<td>418</td>
<td>33DF</td>
<td>1F</td>
<td>OPE</td>
</tr>
<tr>
<td>00000498</td>
<td>STIS</td>
<td>419</td>
<td>33E5</td>
<td>7B</td>
<td>00000499</td>
<td>STIS</td>
<td>420</td>
<td>33E5</td>
<td>79</td>
<td>BAR</td>
</tr>
<tr>
<td>0000045C</td>
<td>STIS</td>
<td>421</td>
<td>33E2</td>
<td>1F</td>
<td>0000045D</td>
<td>STIS</td>
<td>422</td>
<td>33E2</td>
<td>4F</td>
<td>MOV</td>
</tr>
<tr>
<td>0000045F</td>
<td>STIS</td>
<td>423</td>
<td>33E5</td>
<td>7A</td>
<td>00000460</td>
<td>STIS</td>
<td>424</td>
<td>33E5</td>
<td>1F</td>
<td>R1ST FRACTION</td>
</tr>
<tr>
<td>0000045E</td>
<td>STIS</td>
<td>425</td>
<td>33E7</td>
<td>57</td>
<td>00000461</td>
<td>STIS</td>
<td>426</td>
<td>33E7</td>
<td>7B</td>
<td>MOV</td>
</tr>
<tr>
<td>0000045F</td>
<td>STIS</td>
<td>427</td>
<td>33E8</td>
<td>1F</td>
<td>0000045E</td>
<td>STIS</td>
<td>428</td>
<td>33E8</td>
<td>5F</td>
<td>R1ST FRACTION</td>
</tr>
<tr>
<td>0000045E</td>
<td>STIS</td>
<td>429</td>
<td>33EF</td>
<td>2D</td>
<td>0000045F</td>
<td>STIS</td>
<td>430</td>
<td>33EF</td>
<td>07</td>
<td>DCR</td>
</tr>
<tr>
<td>00000460</td>
<td>STIS</td>
<td>431</td>
<td>33EF</td>
<td>C9</td>
<td>00000461</td>
<td>STIS</td>
<td>432</td>
<td>33EF</td>
<td>2D</td>
<td>RET</td>
</tr>
<tr>
<td>00000464</td>
<td>STIS</td>
<td>433</td>
<td>33F1</td>
<td>80</td>
<td>00000465</td>
<td>STIS</td>
<td>434</td>
<td>33F1</td>
<td>7E</td>
<td>MOV</td>
</tr>
<tr>
<td>00000466</td>
<td>STIS</td>
<td>435</td>
<td>33F1</td>
<td>7F</td>
<td>00000466</td>
<td>STIS</td>
<td>436</td>
<td>33F3</td>
<td>77</td>
<td>MOV</td>
</tr>
<tr>
<td>00000467</td>
<td>STIS</td>
<td>437</td>
<td>33F4</td>
<td>AP</td>
<td>00000468</td>
<td>STIS</td>
<td>438</td>
<td>33F5</td>
<td>6F</td>
<td>MOV</td>
</tr>
<tr>
<td>00000469</td>
<td>STIS</td>
<td>439</td>
<td>33F6</td>
<td>93</td>
<td>0000046A</td>
<td>STIS</td>
<td>440</td>
<td>33F7</td>
<td>5F</td>
<td>MOV</td>
</tr>
<tr>
<td>0000046B</td>
<td>STIS</td>
<td>441</td>
<td>33F7</td>
<td>8D</td>
<td>0000046C</td>
<td>STIS</td>
<td>442</td>
<td>33F9</td>
<td>9A</td>
<td>MOV</td>
</tr>
<tr>
<td>0000046D</td>
<td>STIS</td>
<td>443</td>
<td>33FA</td>
<td>57</td>
<td>0000046E</td>
<td>STIS</td>
<td>444</td>
<td>33FB</td>
<td>7D</td>
<td>MOV</td>
</tr>
<tr>
<td>0000046F</td>
<td>STIS</td>
<td>445</td>
<td>33FC</td>
<td>99</td>
<td>00000470</td>
<td>STIS</td>
<td>446</td>
<td>33FD</td>
<td>6F</td>
<td>MOV</td>
</tr>
<tr>
<td>00000471</td>
<td>STIS</td>
<td>447</td>
<td>33FF</td>
<td>7D</td>
<td>00000472</td>
<td>STIS</td>
<td>448</td>
<td>33FF</td>
<td>96</td>
<td>MOV</td>
</tr>
<tr>
<td>00000473</td>
<td>STIS</td>
<td>449</td>
<td>33FF</td>
<td>47</td>
<td>00000474</td>
<td>STIS</td>
<td>450</td>
<td>3401</td>
<td>C9</td>
<td>MOV</td>
</tr>
<tr>
<td>00000475</td>
<td>STIS</td>
<td>451</td>
<td>3402</td>
<td>20</td>
<td>00000476</td>
<td>STIS</td>
<td>452</td>
<td>3404</td>
<td>7B</td>
<td>MOV</td>
</tr>
<tr>
<td>00000477</td>
<td>STIS</td>
<td>453</td>
<td>3404</td>
<td>78</td>
<td>00000478</td>
<td>STIS</td>
<td>454</td>
<td>3405</td>
<td>A7</td>
<td>MOV</td>
</tr>
<tr>
<td>00000479</td>
<td>STIS</td>
<td>455</td>
<td>3406</td>
<td>22</td>
<td>00000480</td>
<td>STIS</td>
<td>456</td>
<td>3406</td>
<td>C2</td>
<td>MOV</td>
</tr>
<tr>
<td>00000481</td>
<td>STIS</td>
<td>457</td>
<td>3407</td>
<td>3A</td>
<td>00000482</td>
<td>STIS</td>
<td>458</td>
<td>3407</td>
<td>53</td>
<td>MOV</td>
</tr>
<tr>
<td>00000483</td>
<td>STIS</td>
<td>459</td>
<td>340C</td>
<td>5F</td>
<td>00000484</td>
<td>STIS</td>
<td>460</td>
<td>340D</td>
<td>7D</td>
<td>MOV</td>
</tr>
<tr>
<td>00000485</td>
<td>STIS</td>
<td>461</td>
<td>340E</td>
<td>0B</td>
<td>00000486</td>
<td>STIS</td>
<td>462</td>
<td>340E</td>
<td>6F</td>
<td>MOV</td>
</tr>
<tr>
<td>00000487</td>
<td>STIS</td>
<td>463</td>
<td>340F</td>
<td>4A</td>
<td>00000488</td>
<td>STIS</td>
<td>464</td>
<td>340F</td>
<td>C9</td>
<td>MOV</td>
</tr>
<tr>
<td>00000489</td>
<td>STIS</td>
<td>465</td>
<td>3411</td>
<td>04</td>
<td>0000048A</td>
<td>STIS</td>
<td>466</td>
<td>3411</td>
<td>04</td>
<td>MOV</td>
</tr>
<tr>
<td>0000048B</td>
<td>STIS</td>
<td>467</td>
<td>3414</td>
<td>5F</td>
<td>0000048C</td>
<td>STIS</td>
<td>468</td>
<td>3414</td>
<td>32</td>
<td>MOV</td>
</tr>
</tbody>
</table>

COMP: DCR L; TO ADDR ACCUM SIGN

COMP1: XRA A; 256

BORN: MVI L, 0040; MAX NORMALIZING SHIFT

NORM1: MOV A, B ; 1ST FRACTION

BORN2: DCR L; DECREMENT SHIFT COUNT

NOPALIZING SHIFT COUNT

ZERO 4TH FRACTION

1ST FRACTION

2ND FRACTION

COMPLEMENT 4TH FRACTION

COMPLEMENT 3RD FRACTION

COMPLEMENT 2ND FRACTION

COMPLEMENT 1ST FRACTION

RETURN TO CALLER

SET CONTROL BITS

Reduced Shift Count

Restore Contents of E Register

Restore Contents of E Register

Op Original Contents of E

Left Shift E

Restore Contents of E Register
<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>SOURCE</th>
<th>INTEL 8080 ASSEMBLER</th>
<th>PAGE 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>469</td>
<td>3419</td>
<td>7A</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,D ;</td>
<td>ORIGINAL CONTENTS OF D REGISTER</td>
<td></td>
</tr>
<tr>
<td>470</td>
<td>341A</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>RAL ;</td>
<td>LEFT SHIFT D</td>
<td></td>
</tr>
<tr>
<td>471</td>
<td>341B</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td>MOV D,A ;</td>
<td>RESTORE CONTENTS OF D REGISTER</td>
<td></td>
</tr>
<tr>
<td>472</td>
<td>341C</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,C ;</td>
<td>ORIGINAL CONTENTS OF C REGISTER</td>
<td></td>
</tr>
<tr>
<td>473</td>
<td>341D</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>RAL ;</td>
<td>LEFT SHIFT C</td>
<td></td>
</tr>
<tr>
<td>474</td>
<td>341E</td>
<td>4F</td>
<td></td>
<td></td>
<td></td>
<td>MOV C,A ;</td>
<td>RESTORE CONTENTS OF C REGISTER</td>
<td></td>
</tr>
<tr>
<td>475</td>
<td>341F</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,B ;</td>
<td>ORIGINAL CONTENTS OF B REGISTER</td>
<td></td>
</tr>
<tr>
<td>476</td>
<td>3420</td>
<td>8F</td>
<td></td>
<td></td>
<td></td>
<td>ADC A ;</td>
<td>LEFT SHIFT B</td>
<td></td>
</tr>
<tr>
<td>477</td>
<td>3421</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td>MOV B,A ;</td>
<td>RESTORE CONTENTS OF B REGISTER</td>
<td></td>
</tr>
<tr>
<td>478</td>
<td>3422</td>
<td>F2</td>
<td>15</td>
<td>34</td>
<td></td>
<td>NORM3:</td>
<td>JP WORM2 ; IF NOT NORMALIZED</td>
<td></td>
</tr>
<tr>
<td>479</td>
<td>3425</td>
<td>7D</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,L ;</td>
<td>NORMALIZING SHIFT COUNT</td>
<td></td>
</tr>
<tr>
<td>480</td>
<td>3426</td>
<td>D6</td>
<td>20</td>
<td></td>
<td></td>
<td>SUZ 040Q ;</td>
<td>REMOVE EIAS</td>
<td></td>
</tr>
<tr>
<td>481</td>
<td>3428</td>
<td>2E</td>
<td>30</td>
<td></td>
<td></td>
<td>MOV L,ACCE ;</td>
<td>TO ADDR ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>482</td>
<td>342A</td>
<td>B6</td>
<td></td>
<td></td>
<td></td>
<td>ADD H ;</td>
<td>ADJUST ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>483</td>
<td>342B</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td>MOV R,A ;</td>
<td>NEW ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>484</td>
<td>342C</td>
<td>C8</td>
<td></td>
<td></td>
<td></td>
<td>RZ ;</td>
<td>RETURN IF ZERO EXP</td>
<td></td>
</tr>
<tr>
<td>485</td>
<td>342D</td>
<td>1F</td>
<td></td>
<td></td>
<td></td>
<td>RAB ;</td>
<td>BORROW BIT TO SIGN</td>
<td></td>
</tr>
<tr>
<td>486</td>
<td>342E</td>
<td>A7</td>
<td></td>
<td></td>
<td></td>
<td>AMA A ;</td>
<td>SET SIGN TO IIND. UNDERFLOW</td>
<td></td>
</tr>
<tr>
<td>487</td>
<td>342F</td>
<td>C9</td>
<td></td>
<td></td>
<td></td>
<td>RET ;</td>
<td>RETURN TO CALLER</td>
<td></td>
</tr>
<tr>
<td>488</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>; SUBROUTINE TO ROUND THE B, C, D REGISTERS.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>489</td>
<td>3430</td>
<td>2E</td>
<td>30</td>
<td></td>
<td></td>
<td>BOND:</td>
<td>MOV A,3 ; TO ADDR ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>490</td>
<td>3432</td>
<td>7B</td>
<td></td>
<td></td>
<td></td>
<td>AMA A ;</td>
<td>SET CONTROL BITS</td>
<td></td>
</tr>
<tr>
<td>491</td>
<td>3433</td>
<td>A7</td>
<td></td>
<td></td>
<td></td>
<td>MOV E,B ;</td>
<td>ACCUMULATOR EXPONENT</td>
<td></td>
</tr>
<tr>
<td>492</td>
<td>3434</td>
<td>5E</td>
<td></td>
<td></td>
<td></td>
<td>CH FNDR ;</td>
<td>CALL 2ND LEVEL ROUNDER</td>
<td></td>
</tr>
<tr>
<td>493</td>
<td>3435</td>
<td>FC</td>
<td>3F</td>
<td>34</td>
<td></td>
<td>BC ;</td>
<td>IF OVERFLOW</td>
<td></td>
</tr>
<tr>
<td>494</td>
<td>3436</td>
<td>DB</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,B ;</td>
<td>1ST FRACTION</td>
<td></td>
</tr>
<tr>
<td>495</td>
<td>3437</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td>LRM L ;</td>
<td>TO ADDR ACCUM SIGN</td>
<td></td>
</tr>
<tr>
<td>496</td>
<td>343A</td>
<td>2C</td>
<td></td>
<td></td>
<td></td>
<td>XRA M ;</td>
<td>ACCUM SIGN AND 1ST FRCTN</td>
<td></td>
</tr>
<tr>
<td>497</td>
<td>343B</td>
<td>AE</td>
<td></td>
<td></td>
<td></td>
<td>JMP STR1 ;</td>
<td>RETURN THRU STORE SUBR.</td>
<td></td>
</tr>
<tr>
<td>498</td>
<td>343C</td>
<td>C3</td>
<td>3F</td>
<td>32</td>
<td></td>
<td>SECOND LEVEL ROUNING SUBROUTINE.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>499</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FNDR:</td>
<td>INR D ; BOUND 3RD FRACTION</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>343F</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>RNZ ;</td>
<td>RETURN IF NO CARRY</td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>3440</td>
<td>C0</td>
<td></td>
<td></td>
<td></td>
<td>INR C ;</td>
<td>CARRY TO 2ND FRACTION</td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>3441</td>
<td>0C</td>
<td></td>
<td></td>
<td></td>
<td>RNZ ;</td>
<td>RETURN IF NO CARRY</td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>3442</td>
<td>C0</td>
<td></td>
<td></td>
<td></td>
<td>INR B ;</td>
<td>CARRY TO 1ST FRACTION</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>3443</td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td>RNZ ;</td>
<td>RETURN IF NO CARRY</td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>3444</td>
<td>C0</td>
<td></td>
<td></td>
<td></td>
<td>MOV A,E ;</td>
<td>ACCUMULATOR EXPONENT</td>
<td></td>
</tr>
<tr>
<td>506</td>
<td>3445</td>
<td>7B</td>
<td></td>
<td></td>
<td></td>
<td>ADI 1 ;</td>
<td>INCREMENT ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>507</td>
<td>3446</td>
<td>C6</td>
<td>01</td>
<td></td>
<td></td>
<td>MOV E,A ;</td>
<td>NEW ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>508</td>
<td>3448</td>
<td>5F</td>
<td></td>
<td></td>
<td></td>
<td>MVI B,200Q ;</td>
<td>NEW 1ST FRACTION</td>
<td></td>
</tr>
<tr>
<td>509</td>
<td>3449</td>
<td>06</td>
<td>80</td>
<td></td>
<td></td>
<td>MOV H,A ;</td>
<td>NEW ACCUM EXPONENT</td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>344A</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td>RET ;</td>
<td>RETURN TO BOND SUBROUTINE</td>
<td></td>
</tr>
<tr>
<td>511</td>
<td>344C</td>
<td>C9</td>
<td></td>
<td></td>
<td></td>
<td>; FIXED POINT MULTIPLY SUBROUTINE.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MUL:</td>
<td>MVI L,MULP1 ; TO ADDR 1ST MULTIPIICA</td>
<td></td>
</tr>
<tr>
<td>513</td>
<td>344D</td>
<td>2E</td>
<td>09</td>
<td></td>
<td></td>
<td>MOV H,A ;</td>
<td>1ST MULTIPLICAND</td>
<td></td>
</tr>
<tr>
<td>514</td>
<td>344F</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td>MVI L,MULP2 ; TO ADDR 2ND MULTIPLICAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>3450</td>
<td>2E</td>
<td>05</td>
<td></td>
<td></td>
<td>MOV N,D ;</td>
<td>2ND MULTIPLICAND</td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>3452</td>
<td>7D</td>
<td></td>
<td></td>
<td></td>
<td>MVI L,MULP3 ;</td>
<td>TO ADDR 3RD MULTIPLICAND</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>3453</td>
<td>2E</td>
<td>01</td>
<td></td>
<td></td>
<td>MOV N,E ;</td>
<td>3RD MULTIPLICAND</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>3455</td>
<td>73</td>
<td></td>
<td></td>
<td></td>
<td>XRA A ;</td>
<td>CLEAR 6TH PRODUCT</td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>3456</td>
<td>AF</td>
<td></td>
<td></td>
<td></td>
<td>MOV E,A ;</td>
<td>CLEAR 5TH PRODUCT</td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>3457</td>
<td>5F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BULK FLOATING POINT MATH PACK-ARITHMETIC AND UTILITY ROUTINES

LINE ADDR B1 B2 B3 B4 ERROR SOURCE INTEL 8080 ASSEMBLER PAGE 11
521 3459 57  NOV D,A ; CLEAR 4TH PRODUCT ; MULTIPLY BY EACH ACCUMULATOR
522  NOV D,A ; FRACTION IN TURN.
523  NOV L,ACC3 ; TO ADDRESS 3RD FRCTN
524  MVI L,ACC3 ; TO ADDRESS 3RD FRCTN
525  CALL MULX2 ; MULTIPLY BY ACCUM 3RD FRCTN
526  MVI L,ACC2 ; TO ADDRESS 2ND FRCTN
527  CALL MULX1 ; MULTIPLY BY ACCUM 2ND FRCTN
528  MVI L,ACC1 ; TO ADDRESS 1ST FRCTN
529  ; MULTIPLY BY ONE ACCUMULATOR WORD.
530  MULX1: NOV A,D ; 5TH PARTIAL PRODUCT
531  NOV E,C ; 4TH PARTIAL PRODUCT
532  NOV D,B ; 3RD PARTIAL PRODUCT
533  MULX2: NOV B,M ; MULTIPLIER
534  NOV L,A ; 5TH PARTIAL PRODUCT
535  XRA A ; ZERO
536  NOV C,A ; 2ND PARTIAL PRODUCT
537  NOV C,A ; 2ND PARTIAL PRODUCT
538  SUB B ; SET CARRY BIT FOR EXIT FLAG
539  JC MULX3 ; IF MULTIPLIER IS NOT ZERO
540  NOV C,D ; 2ND PARTIAL PRODUCT
541  NOV D,E ; 3RD PARTIAL PRODUCT
542  ; EXIT
543  MULX3: NOV C,A ; 2ND PARTIAL PRODUCT
544  JNC MULX3 ; IF NO CARRY TO 1ST PRODUCT
545  INR B ; ADD CARRY TO 1ST PRODUCT
546  ABA A ; CLEAR CARRY BIT
547  ; LOOP FOR EACH BIT OF MULTIPLIER WORD.
548  MULX3: NOV A,L ; 5TH PARTIAL PRODUCT, EXIT
549  ADC A ; SHIFT EXIT FLAG OUT IF DONE
550  BZ ; EXIT IF MULTIPLICATION DONE
551  NOV L,A ; 5TH PARTIAL PRODUCT, EXIT FLAG
552  NOV A,E ; 4TH PARTIAL PRODUCT
553  RAL ; SHIFT 4TH PARTIAL PRODUCT
554  NOV E,A ; 4TH PARTIAL PRODUCT
555  NOV A,D ; 3RD PARTIAL PRODUCT
556  RAL ; SHIFT 3RD PARTIAL PRODUCT
557  NOV D,A ; 3RD PARTIAL PRODUCT
558  NOV A,C ; 2ND PARTIAL PRODUCT
559  RAL ; SHIFT 2ND PARTIAL PRODUCT
560  NOV C,A ; 2ND PARTIAL PRODUCT
561  NOV A,B ; 1ST PART PROD AND MULTIPLIER
562  RAL ; SHIFT 1ST PROD AND MULTIPLIER
563  NOV B,A ; 1ST PART PROD AND MULTIPLIER
564  JNC MULX3 ; IF NO ADDITION REQURED
565  JMP MULX4 ; TO BAR CODE
566  ; FIXED POINT DIVIDE SUBROUTINE.
567  ; SUBTRACT DIVIDEND FROM ACCUMULATOR TO
568  ; OBTAIN 1ST REMAINDER.
569  ; DIVX: MVI L,ACC3 ; TO ADDRESS ACCUM 3RD F
<table>
<thead>
<tr>
<th>LINE</th>
<th>ADDR</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>ERROR</th>
<th>SOURCE</th>
<th>INTEL 8080</th>
<th>ASSEMBLED</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>574</td>
<td>3833</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>575</td>
<td>3832</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>576</td>
<td>3831</td>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>577</td>
<td>3830</td>
<td>7E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>578</td>
<td>3839</td>
<td>9A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>579</td>
<td>3838</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>580</td>
<td>3837</td>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>581</td>
<td>3836</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>582</td>
<td>3835</td>
<td>99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>583</td>
<td>3834</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>584</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>585</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>586</td>
<td>3833</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>587</td>
<td>3832</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>588</td>
<td>3831</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>589</td>
<td>3830</td>
<td>1F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>590</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>591</td>
<td>3833</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>592</td>
<td>3832</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>593</td>
<td>3831</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>594</td>
<td>3830</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>595</td>
<td>3833</td>
<td>7A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>596</td>
<td>3832</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>597</td>
<td>3831</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>598</td>
<td>3830</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>599</td>
<td>3833</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>600</td>
<td>3832</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>601</td>
<td>3831</td>
<td>1F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>602</td>
<td>3830</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>603</td>
<td>3833</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>604</td>
<td>3832</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>605</td>
<td>3831</td>
<td>1F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>606</td>
<td>3830</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>607</td>
<td>3833</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>608</td>
<td>3832</td>
<td>1F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>609</td>
<td>3831</td>
<td>0F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>610</td>
<td>3830</td>
<td>0E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>611</td>
<td>3833</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>612</td>
<td>3832</td>
<td>1C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>613</td>
<td>3831</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>614</td>
<td>3830</td>
<td>2A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>615</td>
<td>3833</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>616</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>617</td>
<td>3833</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>618</td>
<td>3832</td>
<td>7E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>619</td>
<td>3831</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>620</td>
<td>3830</td>
<td>5E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>621</td>
<td>3833</td>
<td>2C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>622</td>
<td>3832</td>
<td>5E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>623</td>
<td>3831</td>
<td>7E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>624</td>
<td>3830</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>
BULK FLOATING POINT MATH PACK-A-RITHMETIC AND UTILITY ROUTINES

LINE ADDR   B1 B2 B3 B4   ERROR SOURCE INTEL 8080 ASSEMBLER PAGE 13
00000666  STSIN
00000667  STSIN
00000668  STSIN
00000669  STSIN
00000670  STSIN
00000671  STSIN
00000672  STSIN
00000673  STSIN
00000674  STSIN
00000675  STSIN
00000676  STSIN
00000677  STSIN
00000678  STSIN
00000679  STSIN
00000680  STSIN
00000681  STSIN
00000682  STSIN
00000683  STSIN
00000684  STSIN
00000685  STSIN
00000686  STSIN
00000687  STSIN
00000688  STSIN
00000689  STSIN
00000690  STSIN
00000691  STSIN
00000692  STSIN
00000693  STSIN
00000694  STSIN
00000695  STSIN
00000696  STSIN
00000697  STSIN
00000698  STSIN
00000699  STSIN
00000700  STSIN
00000701  STSIN
00000702  STSIN
00000703  STSIN
00000704  STSIN
00000705  STSIN
00000706  STSIN
00000707  STSIN
00000708  STSIN
00000709  STSIN
00000710  STSIN
00000711  STSIN
00000712  STSIN
00000713  STSIN
00000716  STSIN

TOTAL ASSEMBLER ERRORS = 0
EXECUTABLE FLOATING POINT MATH PACK-FORMAT CONVERSION ROUTINES

SOURCE INTEL 8080 ASSEMBLER PAGE 1

ORG 3500H

Title: 8080 BINARY FLOATING POINT SYSTEM

; PROGRAMMER CAL OME
; DATE 26 DECEMBER 1973

; ARITH IS THE BEGINNING ADDRESS OF THE
; ARITHMETIC AND UTILITY PACKAGE.

; SCR IS THE BEGINNING ADDRESS OF THE
; RAM USED AS SCRATCHPAD FOR THE SYSTEM.

; BAR LOCATIONS USED BY THE BINARY
; FLOATING POINT SYSTEM.

; ADDRESSES IN THE ARITHMETIC AND UTILITY
; PACKAGE REFERENCED BY THE FORMAT CONVERSION
; PACKAGE.

; SUBROUTINE TO CONVERT points to floating point format.

PROGRAM:

; 3RD INPUT
; INPUT EXPONENT
; 3RD FRACTION
; VALUE 3RD FRACTION
; VALUE 2ND FRACTION
; VALUE 2ND FRACTION
; VALUE 1ST FRACTION
; VALUE 1ST FRACTION
; VALUE EXPONENT
; TEMPORARY STORAGE
; TEMPORARY STORAGE
; TEMPORARY STORAGE
; OVERFLOW FLAG
; SUBTRACTION FLAG
; CHARACTER STRING BANK
; CHARACTER STRING BANK
; TEMPORARY STORAGE
; TEMPORARY STORAGE
; TEMPORARY STORAGE
; ACCUMULATOR 3RD FRACTION
; ACCUMULATOR 2ND FRACTION
; ACCUMULATOR 1ST FRACTION
; SUBTRACTION FLAG
; OVERFLOW FLAG
; ACCUMULATOR EXponent
; ACCUMULATOR SIGN
; ACCUMULATOR 1ST FRACTION
; ACCUMULATOR 2ND FRACTION
; ACCUMULATOR 3RD FRACTION

; ADDR
; ADD
; MUL
; TST
; ZRO
; ABS
; STR

; ARITH+760
; ARITH+1060
; ARITH+1200
; ARITH+1310
; ARITH+1560
; ARITH+2140
; ARITH+2640
; ARITH+3270
; ARITH+5530
; ARITH+6740
; ARITH+7110
; ARITH+7570

; SUBROUTINE TO CONVERT FROM FIXED
; POINT TO FLOATING POINT FORMAT.

; MOV L,E : INPUT EXponent
; MOV E,D : 4TH INPUT FRACTION
; MOV D,C : 3RD INPUT FRACTION
BULK FLOATING POINT BATH PACK-FORMAT CONVERSION ROUTINES

00000061 SYSIN
00000062 SYSIN
00000063 SYSIN
00000064 SYSIN
00000065 SYSIN
00000066 SYSIN
00000067 SYSIN
00000068 SYSIN
00000069 SYSIN
00000070 SYSIN
00000071 SYSIN
00000072 SYSIN
00000073 SYSIN
00000074 SYSIN
00000075 SYSIN
00000076 SYSIN
00000077 SYSIN
00000078 SYSIN
00000079 SYSIN
00000080 SYSIN
00000081 SYSIN
00000082 SYSIN
00000083 SYSIN
00000084 SYSIN
00000085 SYSIN
00000086 SYSIN
00000087 SYSIN
00000088 SYSIN
00000089 SYSIN
00000090 SYSIN
00000091 SYSIN
00000092 SYSIN
00000093 SYSIN
00000094 SYSIN
00000095 SYSIN
00000096 SYSIN
00000097 SYSIN
00000098 SYSIN
00000099 SYSIN
00000100 SYSIN
00000101 SYSIN
00000102 SYSIN
00000103 SYSIN
00000104 SYSIN
00000105 SYSIN
00000106 SYSIN
00000107 SYSIN
00000108 SYSIN
00000109 SYSIN
00000110 SYSIN
00000111 SYSIN
00000112 SYSIN
00000113 SYSIN

SOURCE INTEL 8080 ASSEMBLER PAGE 2

MOV C,B ; 2ND INPUT FRACTION
MOV B,A ; 1ST INPUT FRACTION
MOV A,L ; INPUT EXPONENT
XRI 2000 ; APPLY EXPONENT BITS
MVI H,SCPB ; TO ADDRESS SCRATCH BANK
MVI L,ACCE ; TO ADDP ACCUM EXPONENT
MOV H,A ; ACCUMULATOR EXPONENT
MOV L,A ; TO ADDD ACCUR SIGN
MOV H,2000 ; SET ACCUR SIGN POSITIVE
IMP L ; TO ADDD ACCUM 1ST FRCTN
MOV A,B ; 1ST INPUT FRACTION
ANA A ; SET SIGN BIT
RCL ; INPUT SIGN TO CARRY
JMP ADD10 ; COMPLETE CONVERSION
SUBRTINE TO CONVET FROM FLOATING
POINT TO FIXED POINT FORMAT.

FIX:  MVI H,SCPB ; TO ADDRESS SCRATCH BANK
MVI L,ACCE ; TO ADDP ACCUM EXPONENT
MOV A,B ; ACCUMULATOR EXPONENT
ANA A ; SET CONTROL BITS
JZ FIX1 ; IF ACCUMULATOR IS ZERO
MOV A,Z ; INPUT EXPONENT
SUB m ; SHIFT COUNT
BC ; RETURN IF ACCUM TOO LARGE
CPI 037Q ; COMPARE TO LARGE SHIFT
JNC FIX1 ; IF ACCUMULATOR TOO SMALL
ADI 1 ; SHIFT COUNT
MVI L,ACCE ; TO ADDD ACCUR SIGN
MOV A,L ; ACCUMULATOR SIGN
ANA A ; SET CONTROL BITS
CP COMP ; COMPLEMENT FRCTN IF NEG
MVI A,1 ; MON-ZERO
ORA B ; SET CONTROL BITS FOR EXIT
MOV A,B ; 1ST RESULT
MOV B,C ; 2ND RESULT
MOV C,D ; 3RD RESULT
MOV D,E ; 4TH RESULT
RET ; RETURN TO CALLER

FIX1:  MVI A,0 ; ZERO
MOV B,A ; ZERO
MOV C,A ; ZERO
MOV D,A ; ZERO
RET ; RETURN TO CALLER
DB 0 ; CHECKSUM WORD
HUFF FLOATING POINT HUFF PACK-FORMAT CONVERSION Routines

LINE ADDR B1 B2 B3 B4 ERROR

SOURCE INTEL 8080 ASSEMBLER PAGE 3

: SUBROUTINE ENTRY POINT.
: INITIALIZE TEMORARY STORAG.

IMP: MOV E,8 ; FIRST CHARACTER OF STRING
      CALL TIAS ; SET CHAR ADDR, PNT FLE, FPR
      IMR E ; TO ADDRESS VALUE SIGN
      MVI 4,200 ; SET VALUE SIGN POSITIVE?
      MVI L,ACCE ; TO ADDR ACCUM EXPONENT
      MOV E,D ; SET ACCUM TO ZERO
      MOV A,E ; FIRST CHARACTER
      CPI 360Q ; COMPARE TO SPACE
      JZ IMP1 ; IF SPACE CHARACTER
      CPI 373Q ; COMPARE CHAR TO PLUS
      JZ IMP1 ; IF PLUS SIGN
      CPI 375Q ; COMPARE CHAR TO MINUS
      JNZ IMP2 ; IF NOT MINS SIGN
      MVI L,THP3 ; TO ADDR VALUE SIGN
      MOV H,D ; SET VALUE SIGN NEGATIVE

IMP1: CALL ANALYZE NEXT CHARACTER IN STRING.

IMP2: MOV A,E ; NEXT CHARACTER
      MVI H,SCHB ; TO ADDRESS SCRATCH BANK
      MVI 4,0 ; DIGIT 2ND WD OR DEC EX
      CPI 376Q ; COMPARE TO DECIMAL POINT
      JZ IMP3 ; IF DECIMAL POINT
      MVI 025Q ; COMPARE TO EXPONENT SIGN
      JZ IMP4 ; IF EXPONENT SIGN
      MVI 13Q ; SET CARRY IF CHAR IS DIGIT
      JNC IMP8 ; IF CHAR IS NOT A DIGIT
      MVI L,THP8 ; TO ADDR CURRENT DIGIT
      MOV H,A ; SAVE CURRENT DIGIT
      LIX H,FTP ; TO ADDR FLOATING TEN
      CALL SOL ; MULTIPLY BY TEN
      MVI L,VAL ; TO ADD VALUE
      CALL STR ; STORE OLD VALUE TIMES TEN
      MOV A,9 ; CURRENT DIGIT
      MVI E,0 ; CLEAR 2ND WORD OF DIGIT
      MOV C,B ; CLEAR 2ND WORD OF DIGIT
      MOV D,B ; CLEAR 3RD WORD OF DIGIT
      MVI E,010Q ; INDICATE DIGIT IS IN REG A
      CALL PLT ; CONVERT DIGIT TO FLOATING PNT
      MVI L,VAL ; TO ADD VALUE
      CALL ED ; ADD OLD VALUE TIMES TEN
      MOV A,4 ; DECIMAL POINT FLAG
      AWA A ; SET CONTROL BITS
      JZ IMP1 ; IF NO DEC PNT ENCOUNTERED
      MVI A,9 ; CURRENT DIGIT
      MOV B,N ; TO ADDR INPUT EXPONENT
      MOV B,B ; INPUT EXPONENT
      MVI A,8 ; DECREMENT INPUT EXPONENT
      MOV H,B ; UPDATE INPUT EXPONENT
      JBP IMP1 ; TO GET NEXT CHARACTER


BYPF Floating Point Math Pack-Format Conversion Routines

LINE ADDR B1 B2 B3 B4 ERROR SOURCE INTEL 8080 ASSEMBLER PAGE 4

157 35AB 2E 39 3F ERRP IMPS: 
158 35AD 3A 12 0
159 35AB 37 12 0
160 35AF C2 60 35 
161 35B2 C3 E6 35
162 
163 35B5 CD E2 36
164 35B5 78
165 35B9 47 12 0
166 35BA D6 FD
167 35B5 5F
168 35BD C0 36 35
169 35C0 E0 02
170 35C2 38
171 35C3 C2 06 35
172 35C6 2C
173 35C7 7E
174 35C8 06 00
175 35CA F0 0A
176 35CC D0 00 35
177 35CF 47
178 35D0 2C
179 35D1 7E
180 35D2 FE 0A
181 35D4 B2 D5 35
182 
183 35D7 47
184 35D8 78
185 35D9 C7
186 35DA 87
187 35DB 80
188 35DC 87
189 35DD 01
190 35DE 07
191 35DF 7B
192 35E0 A7
193 35E1 C2 E6 35
194 35E4 90
195 35E5 47
196 35E6 A2 22
197 35E8 2E 3A
198 35EA 4E
199 35EB 82 31
200 35ED 71
201 35EE 78
202 
203 35F2 2E 38
204 35F1 36
205 35F2 CA 59 32
206 35F5 77
207 35F6 21 0D 36
208 35F9 F2 04 36

; PROCESS DECIMAL EXPONENT.

IMPS: 

IMPS: CALL ; CALL CHAR ADDP SBP

IMPS: MOV ; NEXT CHARACTER OF STRING

IMPS: MOV ; CURRENT CHARACTER

IMPS: MOV ; CURRENT CHARACTER

IMPS: MOV ; IF MINS SIGN

IMPS: JZ ; IF MINS SIGN

IMPS: MOV ; IF NOT PLUS SIGN

IMPS: MOV ; NEXT ADDRESS NEXT CHAR

IMPS: MOV ; POSSIBLE DEC EXPONENT

IMPS: CPI ; SET Carry IF CHAR IS DIGIT

IMPS: MOV ; IF CHAR IS NOT A DIGIT

IMPS: MOV ; DEC EX NOT EQUAL DIGIT

IMPS: MOV ; TO ADDRESS NEXT CHAR

IMPS: CPI ; SET Carry IF CHAR IS DIGIT

IMPS: JNP ; IF CHAR IS NOT A DIGIT

IMPS: MOV ; LS DIGIT OF DEC EXP

IMPS: MOV ; MS DIGIT OF DEC EXP

IMPS: ADD ; 2 * MS DIGIT

IMPS: ADD ; 4 * MS DIGIT

IMPS: ADD ; 5 * MS DIGIT

IMPS: ADD ; 10 * MS DIGIT

IMPS: MOV ; DECIMAL EXPONENT

IMPS: MOV ; SIGN OF DEC EXPONENT

IMPS: MOV ; SET CONTROL BITS

IMPS: JNZ ; IF SIGN PLUS

IMPS: MOV ; IF SIGN PLUS

IMPS: MOV ; COMPLEMENT DEC EXP

IMPS: MOV ; DECIMAL EXPONENT

IMPS: MOV ; SCR ; TO ADDRESS SCRATCH BANK

IMPS: MOV ; TO ADDRESS INPUT SIGN

IMPS: MOV ; INPUT SIGN

IMPS: MOV ; TO ADDRESS ACUM SIGN

IMPS: MOV ; ACCUMULATOR SIGN

IMPS: MOV ; DECIMAL EXPONENT

; CONVERT DECIMAL EXPONENT TO BINARY.

IMPS: 

IMPS: MOV ; TO ADDRESS DEC EXPONENT

IMPS: ADD ; ADJUST DECIMAL EXPONENT

IMPS: JZ ; IN DEC EXP IS ZERO

IMPS: MOV ; CURRENT DECIMAL EXPONENT

IMPS: LXI ; TO ADD FLOAT TEN

IMPS: JF ; IF MUTEFLY REQUIRED
RUFF FLOATING POINT RASH PACK-FORMAT CONVERSION ROUTINES

SOURCE INTEL 8080 ASSEMBLER PAGE 5

CALL DIV ; DIVIDE BY TEN
HVI A,1 ; TO INCREMENT DEC EXP
JNP IMP9 ; TO TEST FOR COMPLETION

IMP10: CALL MUL ; MULTIPLY BY TEN
RC ; RETURN IF OVERFLOW
HVI A,777Q ; TO DECREMENT DEC EXP
JNP IMP9 ; TO TEST FOR COMPLETION

OUT SUBROUTINE ENTRY POINT.

OUT CHARACTER ADDRESS AND ACCUMULATOR.

L: DCR L ; DECREMENT CHARACTER ADDRESS
CALL SWAD ; SET CHAR ADDR, DIG CNW, DEC EIP
CALL TST ; LOAD ACCUM TO REGISTERS
HVI L,VALE ; TO ADDR ACCUM SAVE AREA
CALL STD ; CALL REG STR SUBROUTINE

OUT SIGN CHARACTER.
CALL CHAD ; CALL CHAR ADDP SSBTH
HVI H,360Q ; STORE SPACE CHARACTER
ANA A ; SET CONTROL BITS
JZ OUT3 ; IF ACCUMULATOR IS ZERO
HVI A,36Q Q ; ACCUMULATOR EXPONENT
MOV E,A ; ACCUM SIGN AND 1ST FICIENT
ANA A ; SET CONTROL BITS
HVI A,36Q Q ; ACCUMULATOR EXPONENT
JP OUT1 ; IF ACCUM IS POSITIVE
HVI H,37BQ ; CHANGE SIGN TO BCEMS

SCALE ACCUMULATOR TO . . 1. RANGE,

OUT1: CPI 1760 ; COMPARE TO SMALL EXPON
OUT2: LEX H,36H ; ADD TO ADD FLOATING TEN
JC OUT4 ; IF EXPONENT TOO SMALL
CPI 2010 ; COMPARE TO LARGE EXP
JC OUT5 ; IF EXP SMALL ENOUGH
CALL DIV ; DIVIDE BY TEN

OUT3: HVI H,36LB ; TO ADDRESS SCRATCH BAR
HVI L,36PB ; TO ADDR DECIMAL EXPONENT
MOV E,E ; DECIMAL EXPONENT
INP E ; INCREMENT DECIMAL EXPONENT
HVI H,36EB ; DECIMAL EXPONENT
JMP OUT2 ; TO TEST FOR SCALING COMPLETE

OUT4: CALL MUL ; MULTIPLY BY TEN
HVI L,36PB ; TO ADDR DECIMAL EXPONENT
MOV E,E ; DECIMAL EXPONENT
DCB E ; DECREMENT DECIMAL EXPONENT
HVI H,36EB ; DECIMAL EXPONENT
JMP OUT1 ; TO TEST FOR SCALING COMPLETE

FOUND THE VALUE BY ADDING ,000000.

OUT5: CALL ABS ; SET ACCUM POSITIVE
LXI H,3600 ; TO ADDRESS ROUNDER
CALL AD ; ADD THE ROUNDER
CPI 2010 ; CHECK FOR OVERFLOW
JNP 36PS2 ; IF EXP TOO LARGE

SET DIGIT COUNTS.

"T I L,36PB ; TO ADDR DECIMAL EXPONENT"
<table>
<thead>
<tr>
<th>Line</th>
<th>Address</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>Error</th>
<th>Source Intel 8080 Assembler Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>261</td>
<td>3660</td>
<td>7E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A,H ; DECIMAL EXPONENT</td>
</tr>
<tr>
<td>262</td>
<td>3661</td>
<td>5F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV B,A ; DIGITS BEFORE DEC POINT</td>
</tr>
<tr>
<td>263</td>
<td>3662</td>
<td>FE 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPI $00Q ; COMPARE TO LARGE EXP</td>
</tr>
<tr>
<td>264</td>
<td>3664</td>
<td>DA 69 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JRP OUT6 ; IF EXPONENT IN RANGE</td>
</tr>
<tr>
<td>265</td>
<td>3667</td>
<td>1E 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI E,1 ; DIGITS BEFORE DEC POINT</td>
</tr>
<tr>
<td>266</td>
<td>3669</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td>OUT6 :</td>
<td>SUB E ; ADJUST DEC EXPONENT</td>
</tr>
<tr>
<td>267</td>
<td>366A</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV H,A ; DECIMAL EXPONENT</td>
</tr>
<tr>
<td>268</td>
<td>366D</td>
<td>3E 07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI B,7 ; TOTAL NUMBER OF DIGITS</td>
</tr>
<tr>
<td>269</td>
<td>366D</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUB E ; DIGITS AFTER DECIMAL PNT</td>
</tr>
<tr>
<td>270</td>
<td>3662</td>
<td>2C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INR L ; TO ADD 2ND DIGIT CNT</td>
</tr>
<tr>
<td>271</td>
<td>366F</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV H,A ; DIGITS AFTER DECIMAL POINT</td>
</tr>
<tr>
<td>272</td>
<td>3670</td>
<td>1D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DCP E ; DECREMENT DIGIT COUNT</td>
</tr>
<tr>
<td>273</td>
<td>3671</td>
<td>7B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A,F ; DIGITS BEFORE DEC PNT</td>
</tr>
<tr>
<td>274</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>275</td>
<td>3672</td>
<td>2E 38</td>
<td></td>
<td></td>
<td></td>
<td>OUT7 :</td>
<td>VI L,TNP1 ; TO ADD DIGIT CNT</td>
</tr>
<tr>
<td>276</td>
<td>3674</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ADD N ; ADJUST DIGIT COUNT</td>
</tr>
<tr>
<td>277</td>
<td>3675</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV H,A ; NEW DIGIT CNT</td>
</tr>
<tr>
<td>278</td>
<td>3676</td>
<td>FA 93 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JM OUT8 ; IF COUNT PUN OUT</td>
</tr>
<tr>
<td>279</td>
<td>3679</td>
<td>21 BD 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LHI H,FEN ; TO ADD FLOATING TEN</td>
</tr>
<tr>
<td>280</td>
<td>367C</td>
<td>CD 0C 32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL MUL ; MULTIPLY BY TEN</td>
</tr>
<tr>
<td>281</td>
<td>367F</td>
<td>12 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL E,10Q ; TO PLACE DIGIT IN REGA</td>
</tr>
<tr>
<td>282</td>
<td>3681</td>
<td>CD 17 35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL FIX ; CONVERT TO FIXED FORMAT</td>
</tr>
<tr>
<td>283</td>
<td>3684</td>
<td>CD E2 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL CRA ; CALL CHAP ADDR SBRTN</td>
</tr>
<tr>
<td>284</td>
<td>3687</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV S,A ; OUTPUT DECIMAL DIGIT</td>
</tr>
<tr>
<td>285</td>
<td>3680</td>
<td>AF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XRA A ; CLEAR CURRENT DIGIT</td>
</tr>
<tr>
<td>286</td>
<td>3689</td>
<td>1E 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV E,010Q ; BINARY SCALING COR</td>
</tr>
<tr>
<td>287</td>
<td>3689</td>
<td>CD 00 35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL FLT ; RESTORE VALUE MINUS DIGIT</td>
</tr>
<tr>
<td>288</td>
<td>368E</td>
<td>3E FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI A,377Q ; TO ADJUST DIGIT CNT</td>
</tr>
<tr>
<td>289</td>
<td>3690</td>
<td>C3 72 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JRP OUT7 ; LOOP FOR NEXT DIGIT</td>
</tr>
<tr>
<td>290</td>
<td>3693</td>
<td>2E 3A</td>
<td></td>
<td></td>
<td></td>
<td>OUT8 :</td>
<td>VI L,TNP3 ; TO ADD 2ND DIGIT CNT</td>
</tr>
<tr>
<td>291</td>
<td>3695</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV A,H ; DIGITS AFTER DECIMAL PNT</td>
</tr>
<tr>
<td>292</td>
<td>3696</td>
<td>36 FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI A,377Q ; SET 2ND COUNT REG</td>
</tr>
<tr>
<td>293</td>
<td>3698</td>
<td>A7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANA A ; SET CONTROL BITS</td>
</tr>
<tr>
<td>294</td>
<td>3699</td>
<td>FA A6 3b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JM OUT9 ; IF 2ND COUNT RAN OUT</td>
</tr>
<tr>
<td>295</td>
<td>369C</td>
<td>CD E2 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CALL CHAE ; CALL CHAP ADDR SBRTN</td>
</tr>
<tr>
<td>296</td>
<td>368F</td>
<td>26 PE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI E,376Q ; STORE DECIMAL POINT</td>
</tr>
<tr>
<td>297</td>
<td>368A</td>
<td>26 22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI S,SCRB ; TO ADDRESS SCRTN Bnk</td>
</tr>
<tr>
<td>298</td>
<td>3683</td>
<td>C3 72 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JRP OUT7 ; LOOP FOR NEXT DIGIT</td>
</tr>
<tr>
<td>299</td>
<td>3686</td>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td>OUT9 :</td>
<td>DCB L ; TO ADDR DECIMAL EXP</td>
</tr>
<tr>
<td>300</td>
<td>3687</td>
<td>A6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANA M ; DECIMAL EXPONENT</td>
</tr>
<tr>
<td>301</td>
<td>3688</td>
<td>C1 CD 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JZ OUT13 ; IF DECIMAL EXP IS ZERO</td>
</tr>
<tr>
<td>302</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>303</td>
<td>36A8</td>
<td>06 FB</td>
<td></td>
<td></td>
<td></td>
<td>OUT10 :</td>
<td>HVI B,373Q ; PLUS CHA CHARACTER</td>
</tr>
<tr>
<td>304</td>
<td>36AD</td>
<td>F2 B5 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JP OUT10 ; IF EXPONENT IS POSITIVE</td>
</tr>
<tr>
<td>305</td>
<td>3680</td>
<td>06 FD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HVI B,375Q ; CHANGE SIGN TO MINUS</td>
</tr>
<tr>
<td>306</td>
<td>3682</td>
<td>4F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV C,A ; NEGATIVE EXPONENT</td>
</tr>
<tr>
<td>307</td>
<td>3683</td>
<td>AF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IRA A ; ZERO</td>
</tr>
<tr>
<td>308</td>
<td>3684</td>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUB C ; COMPLEMENT EXPONENT</td>
</tr>
<tr>
<td>309</td>
<td>3685</td>
<td>0E FF</td>
<td></td>
<td></td>
<td></td>
<td>OUT11 :</td>
<td>MOV D,A ; UNITS DIGIT</td>
</tr>
<tr>
<td>310</td>
<td>3687</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INR C ; INCREMENT TENS DIGIT</td>
</tr>
<tr>
<td>311</td>
<td>3688</td>
<td>0C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUI 01Q ; REDUCE REMAINDER</td>
</tr>
<tr>
<td>312</td>
<td>3689</td>
<td>D6 0A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
; MATH PACK TRANS ROUTINE
; MOVES B TO H AND C TO L THEN JUMPS TO
; MATH PACK

RTW 3/9/76

TST EQU 3259H
STR EQU 323EH
LOD EQU 326EH
ADD0 EQU 32D0H
SUB0 EQU 32D4H
MUL EQU 328CH
DIV EQU 3284H
INP EQU 354BH
OUTPUT EQU 360DH

ORG 31D5H

SET SO BUPS UP TO MATH PACK

; STRT:
MOVE H.B

; ADDT:
JMP ADD0

; SUBT:
JMP SUB0

; MULT:
JMP MUL

; DIVT:
JMP DIV

; INPT:
JMP INP

; OUTT:
JMP OUTPUT

END

TOTAL ASSEMBLER ERRORS = 0
<table>
<thead>
<tr>
<th>No.</th>
<th>Symbol</th>
<th>Address</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>0007</td>
<td>ADD0</td>
</tr>
<tr>
<td>6</td>
<td>D</td>
<td>0002</td>
<td>DIV</td>
</tr>
<tr>
<td>11</td>
<td>INPT</td>
<td>3548</td>
<td>INPT</td>
</tr>
<tr>
<td>16</td>
<td>M</td>
<td>0006</td>
<td>MUL</td>
</tr>
<tr>
<td>21</td>
<td>PSW</td>
<td>0006</td>
<td>SP</td>
</tr>
<tr>
<td>26</td>
<td>SUBT</td>
<td>31E9</td>
<td>TST</td>
</tr>
</tbody>
</table>
/* RULE PROGRAM */
/* BY V.P. ECDOWELL AND D.J. KEEFE AND R.T. WITEK */
/* FOR BOB CONTRACT # RO 252053 */
/* Argonne National Laboratory */
/* NOVEMBER 8, 1976 */
/* COPYRIGHT 1976 */
/* 1 */

This program requires the use of the Intel PLM Compiler to generate 8080 code. Use the following compiler switch settings:
S*: 9215(SysStack Pointer to 23FFB for 1k RAM, SP=32-RAH=2000H,
SH=256 ORG for proper math pack operation, SF, SG, SB=6, SO=7)

/* DATA DECLARATIONS */
DECLARE LIT LITERALLY 'LITERALLY';
DECLARE SP16 LIT '0,0,0,0,0,0,0,0,0,0,1,240';
DECLARE MASK DATA('0,0,0,0,0,0,0,0,0,0,0,1,240');
DECLARE TUE LITERALLY '11111111';
DECLARE FALSE LITERALLY '0';
DECLARE QI LITERALLY '3FB';
DECLARE OUT BYTE;
DECLARE BITS07(8) BYTE; DECLARE BITS04(8) BYTE;
DECLARE (NET1, NET2, NET3)(13) BYTE;
DECLARE (B0, B1, B2, D0, D1, D2) (8) BYTE;
DECLARE LF LITERALLY '0AB', CR LITERALLY 'ODH', BELL LITERALLY '7';
DECLARE (FAILTST, SELDET) BYTE;
DECLARE DETSEL(2) BYTE;
DECLARE (TIMERMINS, TIMERHRSTIT) BYTE;
DECLARE (TIMTIBSMINSRS) BYTE;
DECLARE (TIMHEAD) BYTE;
DECLARE (SANSPLSHOURSSANSMINS) BYTE;
DECLARE (DETHRSDETMINS) (3) BYTE;
DECLARE RULMSBEAD LITERALLY '7';
DECLARE (YRSMONSDAYSHRS, MINS, SECS) BYTE;
DECLARE (INTT) BYTE;
DECLARE (TSTPWRTSTSOL) BYTE;
DECLARE (TSTART, STOP) ADDRESS;
DECLARE TTYSTT LITERALLY 'O', TTYBUF LITERALLY '1';
DECLARE DATEOK DATA('DATE CORRECT (Y/N)', QM);
DECLARE WISH DATA('DO YOU WANT ');
DECLARE ISB2 DATA('RUN (Y/N)', Q);
DECLARE STOPTIME DATA('INPUT TIME TO STOP');
DECLARE TQB DATA('ENTER TIME INTERVAL BETWEEN SAMPLES HOURS-');
DECLARE TQB DATA('MIN-');
DECLARE SHORT DATA(CR, LF, 'TIME TOO SHORT MUST BE >14 MINS', CR, LF);
DECLARE STU DATA('ENTER TIME TO START HOUR-');
DECLARE ALTSCORSFACTOR(13) BYTE;
DECLARE UL(52) BYTE;
DECLARE (YSMNSDSBS, BS) BYTE;
DECLARE (CAL$TESTCALSTEST1, CALTEST2, CALSTIF?) BYTE;
DECLARE (ALTFACTOR1, ALTFACTOR2) (13) BYTE;
/*PROCEDURES*/

TYPEOUT: PROCEDURE (CHAR); /* TYPE A CHARACTER */
DECLARE (CHAR,I) BYTE;
DO WHILE((INPUT(TTYSTT) AND 808)=FALSE); ENDMETHOD;
/* WAIT FOR TTY */
OUTPUT(TTYBUF)=CHAR; /* PRINT THE CHAR */
IF CHAR=CR THEN /* WAIT ON CR FOR 5700*/
  DO I=0 TO 8; CALL TIME(250); END; /* 200 MSEC WAIT*/
END TYPEOUT; /* THATS ALL */

TYPEIN: PROCEDURE BYTE; /* INPUT A CHAR FROM TTY */
DO WHILE((INPUT(TTYSTT) AND 1)=FALSE); END; /* WAIT FOR DATA*/
RETURN INPUT(TTYBUF); /* GET THE CHAR */
END TYPEIN;

TYINOUT: PROCEDURE; /* ECHOS A CHAR TO TTY */
  N=TYPEIN AND 7FH;
  CALL TYPEOUT(N);
END TYINOUT;

ECHO: PROCEDURE BYTE;
  CALL TYINOUT;
  RETURN(N);
END ECHO;

DECLARE CNT BYTE;

IMPSLINE: PROCEDURE (P);
DECLARE P ADDRESS, (T,C BASED P) BYTE;
CNT=0;
DO WHILE((T:=ECHO)<CR);
  C(CNT)=T; CNT=CNT+1; END;
C(CNT)='-'0';
CALL TYPEOUT(LF);
END IMPSLINE;

CRLF: PROCEDURE; /* OUTPUTS A CR-LF TO THE TTY */
  CALL TYPEOUT(ODH);
  CALL TYPEOUT(OAH);
END CRLF;

PRINTPPROMPT: PROCEDURE; /* PRINTS PROMPT CHAR */
  CALL CRLF;
  CALL TYPEOUT('**');
END PRINTPPROMPT;

PACKBCD: PROCEDURE; /* PRODUCES A PACKED BCD BYTE */
  CALL TYINOUT;
  M=W-'0';
  CALL TYINOUT;
  M=SHL(H4)+(N-'0');
END PACKBCD;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

PRINTSTRING: PROCEDURE(NAME, LENGTH); /* TYPES A DATA STRING*/
   DECLARE NAME ADDRESS, (LENGTH, I, CHAR BASED NAME) BYTE;
   DO I=0 TO LENGTH-1;
   CALL TYPEOUT(CHAR(I));
   END;
END PRINTSTRING;

PRINTBCD: PROCEDURE(B); /*PRINT BCD CHAR PAIR*/
   DECLARE B BYTE;
   CALL TYPEOUT(SHR(B, 4)+'0');
   CALL TYPEOUT((B AND OFH)+'0');
END PRINTBCD;

ERROR: PROCEDURE; /* ERROR RESTART PROCEDURE RESTARTS PROGRAM
   AFTER ANY ERROR*/
   DECLARE ERRMSG DATA (CR,LF,BELL,BELL,
   '** ERROR **',BELL,BELL,CR,LF,LF);
   CALL PRINTSTRING(.ERRMSG,LENGTH(ERRMSG));
   GOTO START;
END ERROR;

DELY$1MS: PROCEDURE; /* 1 MS DELAY */
   DISABLE;
   CALL TIME(10);
   ENABLE;
END DELAY$1MS;

DELY$5MS: PROCEDURE; /* 5 MS DELAY */
   DISABLE;
   CALL TIME(48);
   ENABLE;
END DELAY$5MS;

TYPEDATE: PROCEDURE; /* OUTPUTS CLOCK DATA*/
   DECLARE DATE DATA(CR,LF,'TODAY IS ');
   DECLARE TIME DATA(' THE TIME IS ');
   CALL PRINTSTRING(.DATE,LENGTH(DATE));
   CALL PRINTBCD(MONS);
   CALL TYPEOUT('/');
   CALL PRINTBCD(DAYS);
   CALL TYPEOUT('/');
   CALL PRINTBCD(YRS);
   CALL PRINTSTRING(.TIE,LENGTH(TIME));
   CALL PRINTBCD(HRS);
   CALL TYPEOUT(':');
   CALL PRINTBCD(MINS);
   CALL CRLF;
END TYPEDATE;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

/* SET DATE AND TIME FROM KEYBOARD*/
DECLARE SETYEAR DATA(CR,LF,'ENTER YEAR-');
DECLARE SETMONTH DATA(' ENTER MONTH-');
DECLARE SETDAY DATA(' ENTER DAY-');
DECLARE SETHOUR DATA(' ENTER HOUR-');
DECLARE SETMIN DATA(' ENTER MINUTE-');
CHANGEDATE: PROCEDURE;
DISABLE;
CALL PRINTSTRING (.SETYEAR,LENGTH(SETYEAR));
CALL PACKBCD;
YRS=M;
CALL PRINTSTRING (.SETMONTH,LENGTH(SETMONTH));
CALL PACKBCD;
MONS=M;
CALL PRINTSTRING (.SETDAY,LENGTH(SETDAY));
CALL PACKBCD;
DAYS=M;
CALL PRINTSTRING (.SETHOUR,LENGTH(SETHOUR));
CALL PACKBCD;
HRS=M;
CALL PRINTSTRING (.SETMIN,LENGTH(SETMIN));
CALL PACKBCD;
MINS=M;
CALL CRLF;
SECS=0;
OUTPUT(OFBH)=0;
ENABLE;
CALL TYPEDATE;
END CHANGEDATE;

RESTART: PROCEDURE INTERRUPT 0;
/* STARTUP AND RESTART PROCEDURE INT 0*/
GO TO START; /* ALL YOU DO ON INT 0 IS GO TO START*/
END RESTART;
TIMEKEEPER: PROCEDURE INTERRUPT 7;
    /*MACHINE IS INTERRUPTED ONCE PER SEC TO UPDATE CLOCK*/
    /*DAYMAX IS A VECTOR FOR LENGTH OF MONTH*/
    DECLARE DAYMAX DATA(0, 31H, 28H, 31H, 30H, 31H, 30H, 31H, 30H, 31H,
        0, 0, 0, 0, 0, 31H, 30H, 31H);
    TIMERSECS=TIMERSECS+1;
    SECS=SECS OR SECS;/*RESETS CARRY BIT TO ACCOUNT FOR COMPILER BUG*/
    IF(SECS:=DEC(SECS+1))=60H THEN /*INCR SECONDS*/
        DO;
            SECS=0;
            SECS=SECS OR SECS; /*SEE ABOVE*/
            TIMERMINS=DEC(TIMERMINS+1); /*EXTERNAL TIMER*/
            IF(TIMERMINS=60H THEN
                DO;
                    TIMERMINS=0; SECS=SECS OR SECS; TIMERHRS=DEC(TIMERHRS+1); END;
                    MINS=MINS OR MINS; /*SEE ABOVE*/
                    IF(MINS:=DEC(MINS+1))=60H THEN /*INCR MINS*/
                        DO;
                            MINS=0;
                            HRS=HRS OR HRS; /*SEE ABOVE*/
                            IF(HRS:=DEC(HRS+1))=24H THEN /*INCR HRS*/
                                DO;
                                    HRS=0;
                                END;
                                /*NOW INCR DAYS AND THEN CHECK FOR LENGTH OF MONTH*/
                                DAYS=DAYS OR DAYS; /*SEE ABOVE*/
                                IF((DAYS:=DEC(DAYS+1)) > DAYMAX(MONS))
                                    THEN DO;
                                        DAYS=1;
                                        DAYS=DAYS OR DAYS;
                                        IF(MONS:=DEC(MONS+1))=13H THEN
                                            DO;
                                                YRS=DEC(YRS+1);
                                                MONS=1;
                                            END;
                                        END;
                                        END;
                                END;
                                END;
                                END;
    END;
    OUTPUT(OFBH)=0;
END TIMEKEEPER;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

DETHSG: PROCEDURE; /*PRINTS DET NUMBER*/
DECLARE COHMSG DATA(' DETECTOR '):
CALL PRINTSTRING(.COHMSG,LENGTH(COHMSG)):
CALL TYPEOUT(SELDET + '1'):
END DETHSG;

/* CHAN 7 OUTPUT PROCEDURE */
/* OUTPUT 7 CONTROLS THE RWLM HEAD*/
/* BIT 0 HEAD SELECT-0=HEAD 1,1=HEAD 2 */
/*BIT 1 */
/*BIT 2 */
/*BIT 3 MOTOR PULSES */
/*BIT 4 MOTOR POWER */
/*BIT 5 SOLENOID POWER */
/*BIT 6 PUMP ON */
/*BIT 7 SOURCE CHECK */
DETSHEAD$OUT: PROCEDURE:
DECLARE I BYTE;
OUT-FALSE;
DO I-0 TO 7;
CUT-OUT OR (BITS07(I) AND HASK(I));
END;
OUTPUT(7)=OUT;
END DETSHEAD$OUT;

/* CHAN 4 OUTPUT PROCEDURE */
/* OUTPUT 4 CONTROLS THE DATA ACCUMULATORS */
/* BIT 0 STORE(NOT) */
/* BIT 1 SET(NOT) */
/* BIT 2 SCAN */
/* BIT 3 CLEAR(NOT) */
/* BIT 4 COUNT ENABLE */
/* BIT 5 */
/* BIT 6 */
/* BIT 7 */
CHNS4$OUT: PROCEDURE;
DECLARE I BYTE;
OUT-FALSE;
DO I-0 TO 7;
CUT-OUT OR (BITS04(I) AND HASK(I));
END;
OUTPUT(4)=OUT;
END CHNS4$OUT;

RELEASE$SOLENOID: PROCEDURE;
DECLARE SOLMSG DATA(' SOLENOID NOT RELEASED');
FAILTST-FALSE;
BIT$07(5)=FALSE;
CALL DETSHEAD$OUT;
IF (INPUT(RWLM$HEAD) AND HASK(5))=FALSE THEN DO;
CALL DETMSG;
CALL PRINTSTRING(.SOLMSG,LENGTH(SOLMSG)):
FAILTST-TRUE; CALL ERROR;
END;
END RELEASE$SOLENOID;
/PAPADV PROCEDURE:
DECLARE PULSES ADDRESS;
DECLARE FINISH LABEL;
DECLARE NOPAPER DATA('FILTER SUPPLY EXHAUSTED',CP,LF);
DECLARE EXTRASADVANCE DATA('FILTER ADVANCE ERROR',CP,LF):
FAILTST=FALSE;
BIT$07(4)=MASK(4):
CALL RELEASE$SOLENOID;
IF FAILTST=0 THEN
   DO PULSES = 0 TO 1150;
   /\TURN ON SOLENOID WHEN STARTING PAPER TAKE UP*/
   IF PULSES > 500 THEN BIT$07(5)=TRUE;
   /\START FORMING MOTOR PULSE*/
   BIT$07(3)=FALSE;
   CALL DET$HEADSOUT;
   CALL DELAYSIMS;
   BIT$07(3)=MASK(3);
   CALL DET$HEADSOUT;
   CALL DELAYS5MS;
   IF ((INPUT(RWLSHEAD) AND MASK(7))=MASK(7) AND PULSES>100 ) THEN
      DO;
      IF (INPUT(RWLSHEAD) AND MASK(4)) = 0 THEN
         DO; CALL DETRSG;
         CALL PRINTSTRING (.NOPAPERLENGTH(NOPAPER));
         FAILTST=TRUE; /* SET ERROR FLAG */
         END;
         GO TO FINISH; /* HOME NICELY*/
      END;
   END;
   /\IF HERE THERE HAS BEEN AN ERROR*/
   CALL DETRSG;
   CALL PRINTSTRING (.EXTRASADVANCELENGTH(EXTRASADVANCE));
   FAILTST=TRUE;
FINISH:BIT$07(4)=FALSE; /*TURN OFF POWER*/
BIT$07(5)=FALSE; /*AND THE SOLENOID*/
CALL DET$HEADSOUT; /*DO IT NOW*/
   IF FAILTST=TRUE THEN CALL ERROR; /*ERROR IF BADNESS SET*/
END PAPADV; /*ELSE GO BACK TO WHERE YOU CAME FROM*/
/* THIS SECTION INTERFACES PLM TO THE MATH PACK */
DECLARE ADDT LIT '31E2H';
DECLARE SUBT LIT '31E7H';
DECLARE INITT LIT '322FH';
DECLARE MOLT LIT '31ECi';
DECLARE DIVT LIT '31F1H';
DECLARE INPT LIT '31F6H';
DECLARE OUTT LIT '31F8H';
DECLARE STRT LIT '3103H';
DECLARE LODT LIT '31DHH';
INIT: PROCEDURE;
GOTO INITT;
END INIT;
ADD: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GOTO ADDT;
END ADD;
SUB: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GOTO SUBT;
END SUB;
MUL: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GO TO MOLT;
END MUL;
DIV: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GO TO DIVT;
END DIV;
LOD: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GO TO LODT;
END LOD;
STR: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GOTO STRT;
END STR;
INP: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GOTO INPT;
END INP;
OUTPT: PROCEDURE(LOC);
DECLARE LOC ADDRESS;
GO TO OUTT;
END OUTPT;
INSTR: PROCEDURE(A1,A2);
DECLARE (A1,A2) ADDRESS;
CALL INP(A1) CALL STR(A2);
END INSTR;
/* THIS END THE MATH PACK - PLM INTERFACE */
DECLARE TEMP(48) BYTE;
DECLARE ASUB,PSUB,TSUB ADDRESS;
DECLARE ALTCONF(4) BYTE;
DECLARE FT(12) BYTE,FB(24) BYTE, FCON(48) BYTE;

NETSCOUNT:PROCEDURE; /*SUBTRACTS BACKGROUND FROM SAMPLE*/
    DO I=0 TO 2;
        ASUB=.FB+I*8;PSUB=ASUB+12;TSUB=.FT+I*4;
        CALL LOD(PSUB) ;CALL SUB(ASUB) ;CALL STR(TSUB);
        CALL OUTPT(.NET1+I*13);
    END;
END NETSCOUNT;

CAL: PROCEDURE; /*PERFORMS CALCULATIONS*/
    DECLARE (I,I1) BYTE;
    CALL INIT;
    /*CONVERT CONSTANT VALUES TO FLOATING POINT AND STORE*/
    DO I=0 TO 11;
        CALL INSTR(.CONSTR+(I*16)+(192*SELDET),.FCON+(I*8));
    END;
    /*CONVERT ALTITUDE CORRECTION FACTOR TO FP AND STORE*/
    /*DETERMINE WHICH HEAD IS SELECTED*/
    IF SELDET=0 THEN
        DO I=0 TO 13;ALTSCORSFACTOR(I)=ALTFACTOR1(I);END;
    ELSE
        DO I=0 TO 13;ALTSCORSFACTOR(I)=ALTFACTOR2(I);END;
    CALL INSTR(.ALTSCORSFACTOR,.ALTCONF);
    /*CONVERT BACKGROUND AND SAMPLE COUNTS TO FP AND STORE*/
    DO I=0 TO 5;
        CALL INSTR(.B0+I*8,.FB+I*4);
    END;
    /*SUBTRACT BACKGROUND FROM SAMPLE AND STORE*/
    CALL NETSCOUNT;
    /*PERFORM MULTIPLICATIONS AND STORE RESULTS IN TEMP*/
    DO I1=0 TO 3 :
        DO I=0 TO 8 BY 4;
            CALL LOD(.FT+I);
            CALL MUL(.FCON+I+(I1*12));
            CALL STR(.TEMP+I+(I1*12));
        END;
        CALL LOD(.TEMP+(I1*12));
        DO I=4 TO 8 BY 4;
            CALL ADD(.TEMP+I+(I1*12));
        END;
    /*PERFORM ADDITIONS*/
    CALL ADD(.TEMP+(I1*12));
    END;
    /*MULTIPLY BY ALTITUDE CORRECTION FACTOR*/
    CALL MUL(.ALTCONF);
    /*STORE RESULTS*/
    CALL OUTPT(.WL+(I1*13));
    END;
END CAL;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

/* THIS SECTION PRINTS THE REPORT ON THE TERMINAL */
DECLARE WLM DATA ("WORKING LEVEL RADIUS A RADIUS B RADIUS C");
DECLARE L DATA ('PC/L PC/L PC/L');
DECLARE HEAD1 DATA (CR,LF,LF,'*** R W L ***',CR,LF);
DECLARE CL DATA ('RADIUH A RADIUM (B+C) RADIUM C");
DECLARE CH DATA ('SAMPLE COUNTS', BACKGROUND COUNTS NET COUNTS',CR,LF);
DECLARE ALTHSG DATA (CR,LF,'ENTER ALTITUDE CORRECTION FACTOR = ');

PRINTREPORT: PROCEDURE;
DECLARE I BYTE;
/* CONVER COUNT DATA FROM BCD TO ASCII*/
DO I=0 TO 47; B0(I)=B0(I)+'0'; END;
DO I=0 TO 38; NET1(I)=NET1(I)+'0'; END;
/* CHECK IF CALIBRATION REPORT*/
IF CALSTEST2=TRUE THEN GOTO CALSPRINT;
DO I=0 TO 51; WL(I)=L(I)+'0'; END;
CALL PRINTSTRING (.HEAD1,LENGTH(HEAD1));
CALL DETKSG;
CALL CRLF;
CALL TYPEDATE;
CALL PRINTSTRING (.ALTHSG+8,27);
DO I=0 TO CNT-1; N=ALTSCORSFACTOR(I)+30H; CALL TYPEOUT(N); END;
CALL CRLF; CALL CRLF;
DO I=0 TO 3;
CALL PRINTSTRING (.WLH+I*24,24);
CALL PRINTSTRING (.WL+I*13,13);
CALL PRINTSTRING (.L+I*5,5);
CALL CRLF;
END;
CALL CRLF; CALL CRLF; CALL CRLF;
/* PRINT CALIBRATION REPORT*/
CALSPRINT:;
CALL PRINTSTRING (.CHLENGTH(CH));
DO I=0 TO 2;
CALL PRINTSTRING (.CH+I*16,16);
CALL PRINTSTRING (.CL+I*16,16);
CALL PRINTSTRING (.BO+I*8+24,7);
CALL PRINTSTRING (.BO+I*8+7,7);
CALL PRINTSTRING (.HEAD1+3,16);
CALL PRINTSTRING (.HEAD1+3,16);
CALL PRINTSTRING (.NET1+I*13,13);
CALL CRLF;
END;
CALL CRLF; CALL CRLF;
/* RETURN DATA TO ORIGINAL CONDITION*/
/* RECONVERT TO BCD*/
DO I=0 TO 47; B0(I)=B0(I)-'0'; END;
DO I=0 TO 38; NET1(I)=NET1(I)-'0'; END;
DO I=0 TO 51; WL(I)=WL(I)-'0'; END;
END PRINTREPORT;
COUNT: PROCEDURE;
/* TURN COUNTERS ON AT TSTART AND OFF
AT TSTOP, CLEAR BEFORE COUNT */

IF CALSTEST1=TRUE THEN GO TO CAL$LOOP;

BIT$04(3)=FALSE; BIT$04(4)=FALSE; CALL CHNS4$OUT;
BIT$04(3)=MASK(3); BIT$04(4)=MASK(4);
DO WHILE (TIMERSEC<TSTART); END;
CALL CHNS4$OUT;

DECLARE TS ADDRESS;
STARTTIM: DISABLE;
TS= TIMERSEC; ENABLE;
IF TSTOP>TS THEN GOTO STARTTIM;
CALL CHNS4$OUT;
END COUNT;

CAL$LOOP:;
BIT$04(4)=FALSE;

DECLAR TS ADDRESS;
STARTTIM: DISABLE;
TS= TIMERSEC; ENABLE;
IF TSTOP>TS THEN GOTO STARTTIM;
CALL CHNS4$OUT;
END COUNT;
READSACCUM: PROCEDURE(A,B,C);
DECLARE I BYTE;

/* READ ACCUM PROCEDURE*/
WILL READ DATA FROM SELECTED HEAD AND RETURN DATA */
DECLARE (A,B,C) ADDRESS, (R1 BASED A, R2 BASED B, R3 BASED C) BYTE;

/* LATCH LSD*/
BITS04 (0) = MASK (0); CALL CHNS4SOUT;

BITS04(0)=FALSE; CALL CHNS4SOUT;

/* INPUT LSD AND RIGHT JUSTIFY */
R1(6) = SHR (INPUT(4),4);
R2(6) = SHR (INPUT(5),4);
R3(6) = SHR (INPUT(6),4);

/* SET THE COUNTER DISPLAY COUNTER*/

BITSO4 (1) = MASK (1); CALL CHNS4SOUT;

BITSO4(1)=FALSE; CALL CHNS4SOUT;

/* THIS LOOP READS THE SIX DIGITS FROM THE COUNTERS*/
DO I=0 TO 5;

/* LATCH THE DIGIT */
BITSO4 (0) = MASK (0); CALL CHNS4SOUT;

BITSO4(0)=FALSE; CALL CHNS4SOUT;

/* INPUT DIGIT AND STRIP OFF LSD*/
R1(I) = NOT (INPUT(4)) AND 15;
R2(I) = NOT (INPUT(5)) AND 15;
R3(I) = NOT (INPUT(6)) AND 15;

/* STROBE THE SCAN INPUT HAS TO BE SLOW FOR THE CMOS*/
BITSO4(2) = MASK(2); CALL CHNS4SOUT;

CALL DELAYS1NS;

BITSO4(2) = FALSE; CALL CHNS4SOUT;

CALL DELAYS1NS;
END;

/* PUT $0 ON THE END OF EACH NUMBER FOR THE MATH PACK*/
R1(7) = '$' - '0'; R2(7) = '$' - '0'; R3(7) = '$' - '0';
END READSACCUM;

DECLARE FLOSHSG DATA(' FLOW RATE OUT OF RANGE '):
FLOWSCHECK: PROCEDURE;
IF (INPUT(RULESHEAD) AND MASK(3)) =0 THEN
DO;
CALL DETSHEADSOUT; /* STOP PUMP*/
CALL DELPHG;
CALL PRINTSTRING (.FLOSHSG_LENGTH(FLOSHSG));
CALL ERROR;
END;
END FLOWSCHECK;
TESTSACSPOWER: PROCEDURE;
DECLARE PWRMSG DATA(' POWER FAIL',CR,LF):
FAILST = FALSE;
IF (INPUT(BWLSHEAD) AND MASK(6)) = 0 THEN
  DO;
    CALL DETMSG;
    CALL PRINTSTRING(.PWRMSG, LENGTH(PWRMSG)); /* POWER FAILURE*/
    FAILST = TRUE;
  END;
END TESTSACSPOWER;

SAMPLE: PROCEDURE;
/* SAMPLE PROCEDURE TAKES SAMPLE FROM HEAD
  HEAD SHOULD BE SELECTED BEFORE CALL*/
/* MAKE SURE THE POWER IS ON SO THAT THINGS WILL RUN*/
CALL TESTSACSPOWER; IF FAILST THEN CALL ERROR;
/* TOP OUT THE PAPER BEFORE YOU START TO WORK*/
/* CHECK FOR ZERO POSITION*/
IF (INPUT(BWLSHEAD) AND MASK(7)) = FALSE THEN CALL PAPADV;
CALL PAPADV; CALL PAPADV;
/* CLEAR OUT PAPER */
TIMERSSECS=0; /* SET REAL TIME TO ZERO*/
/* TAKE BACKGROUND COUNT FROM 1 SEC TO 121 SEC*/
TSTART=1; TSTOP=121; CALL COUNT;
/* READ IN BACKGROUND COUNT*/
CALL READSACCUM(.BO,.B1,.B2);
/* SET TO START PUMP IN OUTPUT VECTOR*/
BIT$07(5)=TRUE; BIT$07(6)=TRUE;
/* WAIT TILL TIME TO START PUMP*/
DO WHILE (TIMERSSECS<122); END;
/* START PUMP*/
CALL DETSHDOUT; BIT$07(6)=FALSE;
/* TEST FLOW RATE AT 200 SEC*/
DO WHILE (TIMERSSECS<200): END;
/* LET PUMP RUN FOR 2 MIN*/
DO WHILE (TIMERSSECS<242): END;
/* STOP PUMP*/
CALL DETSHDOUT;
/* MOVE SAMPLE UNDER HEAD*/
CALL PAPADV;
/* SET UP TO COUNT SAMPLE*/
TSTART=256; TSTOP=376; CALL COUNT;
/* READ DATA FOR SAMPLE*/
CALL READSACCUM(.DO,.D1,.D2);
/* WHEN CALIBRATING JUMP BACK TO CALIBRATE PROCEDURE*/
IF CALSTEST=TRUE THEN GOTO FINISH;
/* DO MATH AND PRINT IT OUT*/
CALL CAL; CALL PRINTREPORT;
/* GO ON HOME NOW ALL IS DONE*/
FINISH:
END SAMPLE;
CYCLE: PROCEDURE;
CALSTEST=FALSE; /*NORMAL RUN*/
/* WAIT TILL TIME TO START*/
DO WHILE (SHIN<>HINS OR SHRS<>HRS):END;
/*ONCE IN INTERRUPT ONLY WAY OUT */
TT=FALSE;
DO WHILE NOT(TT);/*MAKE A STATEMENT AFTER TEST*/
/* CLEAR MIN AND HRS TIMER*/
TIMERSHINS=0; TIMERSHRS=0;
/*HEAD 0 SELECT*/
IF READ(0) THEN
DO;
   BITS07(0)=FALSE;SELDET=G;CALL DET$HEAD$OUT;CALL SAMPLE;
   DO WHILE (TIMERSHINS<07H):END;
   END;
IF READ(1) THEN
   DO;
      BITS07(0)=TRUE;SELDET=1;CALL DET$HEAD$OUT;CALL SAMPLE;
   END;
   DO WHILE NOT(TIMERSHINS=TIM AND TIMERSHRS=TIM):END;
TT=(YS>=YS AND HINS>=HS AND DAYS>=DS AND HRS>=HS AND HINS>=HS); END;
END CYCLE; /*THATS ALL THERE IS*/
MAIN: /* THE ONE THAT STARTS IT OFF*/
CALL TYPEDATE; CALL PRINTSTRING (.DATEOK, LENGTH (.DATEOK));
/* IF DATE OR THEN JUST TURN ON CLOCK ELSE CHANGE DATE*/
/* PRINT DATE AND SEE IF IT IS OK*/
CALL TYINOUT; IF N='N' THEN CALL CHANGEDATE;
ELSE DO: OUTPUT (OFBH) = O; FNABLE; CALL CRLF; END;
DO SELDET = 0 TO 1; /* ASK FOR EACH HEAD*/
CALL PRINTSTRING (.NISH, LENGTH (.WISH)); CALL DETMSG;
CALL PRINTSTRING (.WISH2, LENGTH (.WISH2)); CALL TYINOUT;
IF N='Y' THEN HEAD (SELDET) = TRUE;
ELSE HEAD (SELDET) = FALSE;
CALL CRLF;
TIT = TRUE; DO WHILE (TIT = TRUE);
CALL CRLF; CALL PRINTSTRING (.TQH, LENGTH (.TQH));
CALL PACKBCD; TIH = N;
CALL PRINTSTRING (.TQM, LENGTH (.TQM));
CALL PACKBCD; TIM = N;
CALL CRLF; IF ((TIM < 1 AND TIM < 14H) OR TIM > 59H) THEN TIT = TRUE;
ELSE TIT = FALSE;
IF TIT THEN CALL PRINTSTRING (.SHORT, LENGTH (.SHORT));
END;
CALL PRINTSTRING (.STM, LENGTH (.STM)); CALL PACKBCD;
SHRS = N; CALL PRINTSTRING (.STM, LENGTH (.STM)); CALL PACKBCD;
SHMS = N; CALL CRLF;
CALL PRINTSTRING (.STOPTIME, LENGTH (.STOPTIME));
CALL PRINTSTRING (.SETYEAR, LENGTH (.SETYEAR));
CALL PACKBCD; YS = N;
CALL PRINTSTRING (.SETMON, LENGTH (.SETMON));
CALL PACKBCD; MMS = N;
CALL PRINTSTRING (.SETDAY, LENGTH (.SETDAY));
CALL PACKBCD; DDS = N;
CALL PRINTSTRING (.SETHR, LENGTH (.SETHR));
CALL PACKBCD; HHS = N;
CALL PRINTSTRING (.SETHM, LENGTH (.SETHM));
CALL PACKBCD; HHS = N;
CALL CRLF;
CALL CYCLE;
END MAIN;

DECLARE LST DATA (CR, LF, 'ILLEGAL CHAR PLEASE REENTER', CR, LF);
DECLARE SELECTHEAD DATA(CR,LF,'SELECT HEAD 1 OR 2-');
SELECTHEAD:PROCEDURE; /*SETS UP READ BITS FROM INPUT DATA*/
CALL PRINTSTRING(.SELECTHEAD,LENGTH(SELECTHEAD));
CALL TIMOUT;CALL CRFL;
N=N-'O';IF N=0 THEN GOTO EXIT;IF N>2 THEN GOTO EXIT;
IF N=1 THEN DO;
  SELDET=0;BITS07(0)=FALSE; END;
ELSE DO;
  SELDET=1;BITS07(0)=TRUE;END;
GOTO FINISH;
DECLARE NPOSS DATA(CR,'NOT POSSIBLE');
EZIT:CALL PRINTSTRING(.NPOSS,LENGTH(NPOSS));
GOTO LOOP;
FINISH:;
END SELECTHEAD;

RSTARTSACCU:PROCEDURE; /*RESTARTS ACCUM AFTER READ*/
  CALL CHNS4SOUT;
END RSTARTSACCUB;

/*THIS PROCEDURE PRINTS THE CALIBRATION DATA*/
PRINTSCALDATA:PROCEDURE;
DECLARE CALSHEAD DATA(' MINUTE COUNT',CR,LF,LF);
CALL PRINTSTRING(.BEAD1+3,20);CALL PRINTBCD(CALSTIME);
CALL PRINTSTRING(.CALSHEAD,LENGTH(CALSHEAD));
CALL PRINTREPORT;
END PRINTSCALDATA;

/*THIS PROCEDURE ADJUSTS THE 2 MIN BACKGROUND FOR THE CALIBRATION RUN REQUIREMENTS*/
NETSBKG:PROCEDURE;
CALL CAL1;
DECLARE MLP1 DATA(2,0FEH,5,0,0,24H);
DECLARE MLP2 DATA(1,5,0FEH,0,0,24H);
DECLARE MLP3 DATA(1,7,0FEH,5,0,24H);
DECLARE MLP4 DATA(1,0FEH,0,0,0,24H);
DO I=0 TO 2;
  CALL INP(.KLP1+*6);CALL KUL(.FB+I*4);CALL STP (.FB+I*4);
END;
CALL NETSCOUNT;
END NETSBKG;

CALLSACCUB:PROCEDURE; /*SUBROUTINE FOR CALIBRATE PROCEDURE*/
CALL CCUNT; CALL READSACCU(.DO,.D1,.D2); CALL RSTAPTSACCUB;
/*CALCULATE AND PRINT*/
CALL NETSBKG; CALL PRINTSCALDATA;
END CALLSUB;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

CALIBRATE: PROCEDURE; /* CONTROLS CALIBRATION CYCLE */
DECLARE(N1,N2) BYTE;
DO N1 = 1 TO 2;
IF N1 = 1 THEN DO; SELDET = 0; BIT$07(0) = FALSE; END;
ELSE DO; SELDET = 1; BIT$07(0) = TRUE; END;
DO N2 = 0 TO 9;
OUTPUT(OFBH) = 0; ENABLE; /* BE SURE CLOCK IS RUNNING */
CALSTEST = TRUE;
CALL SAMPLE; /* TAKE A NORMAL RUN */
CALL BSTART$ACCUM;
/* DO MATH AND PRINT IT OUT */
CALL CAL; CALL PRINT$REPORT;
CALSTEST1 = TRUE; CALSTEST2 = TRUE;
/* SET UP TO GET 5 MIN DATA */
CAL$TIME = 05H;
TSTOP = 556; M = 0; CALL CAL$SUB;
/* SET UP TO GET 30 MIN DATA */
CAL$TIME = 30H;
TSTOP = 2056; M = 1; CALL CAL$SUB;
/* SET UP FOR 35 MIN DATA */
CAL$TIME = 35H;
TSTOP = 2356; M = 2; CALL CAL$SUB;
/* SET UP TO DO KUSNETZ METHOD */
M = 3;
CAL$TIME = 39H;
CALSTEST1 = FALSE;
CALSTEST = FALSE; TSTART = 2596; TSTOP = 2716;
CALL CAL$SUB;
CALSTEST2 = FALSE;
IF INPUT(TTYBUF) AND 7FH = 1BH THEN GOTO FINISH;
END; /* END OF INNER LOOP - 10 CAL PUNS */
END; /* END OF OUTER LOOP - BOTH HEADS */
FINISH; /* ESCAPE ROUTE */
END CALIBRATE;
SOURCE$CHECK: PROCEDURE; /*ALLOWS CALIBRATION WITH SOURCES*/ DECLARE SCHK DATA('SOURCE CHECK MODE ');
DECLARE SELS$SOURCE DATA(CR,LF,'SELECT MODE A-B-C -'); DECLARE I BYTE;
START: CALL SELS$HEAD; CALL DET$HEAD$OUT;
  CALL TESTSAC$POWER;
  IF FAILURE=TRUE THEN GOTO LOOP;
  CALL PRINTSTRING (.SELS$SOURCE, LENGTH(SELS$SOURCE));
  CALL TIMEOUT; CALL CRFL;
  /* CHECK ALLOWABLE VALUES OF INPUT*/
  IF N<'A' THEN GOTO EXIT; IF N>'C' THEN GOTO EXIT;
  /*CHECK MODE AND SET UP OUTPUT PORTS*/
  IF N='C' THEN BITS07(7)=TRUE;
  ELSE BITS07(7)=FALSE; CALL DET$HEAD$OUT;
  /*TAKE A TWO MINUTE COUNT*/
  OUTPUT(OFPB)=0; ENABLE; /*BE SURE CLOCK IS RUNNING*/
  TIMERS$SECS=0; TSTART=1; TSTOP=121; CALL COUNT; /*GET DATA*/
  /* READ COUNT AND STORE*/
  CALL READSACCUM (.DO,.D1,.D2);
  /*ZERO UNSELECTED DATA AREAS*/
  DO CASE N='A';
  DO; /* CASE 0, N=A, ZERO (B+C) AND C*/
  DO I=0 TO 6;
  D1(I)=0; D2(I)=0; END;
  END;
  END;
  DO; /* CASE 1, N=B, ZERO A AND C*/
  DO I=0 TO 6;
  D1(I)=0; D2(I)=0; END;
  END;
  END; /*CASE 2, N=C, ZERO A AND (B+C)*/
  DO I=0 TO 6;
  D1(I)=0; D2(I)=0; END;
  END; /*END CASE*/
  /* ZERO BACKGROUND DATA AREA*/
  DO I=0 TO 23; BO(I)=0; END;
  BO(7)=24H; B1(7)=24H; B2(7)=24H; /*PLACE $ FOR MATH PACK DELIMITER*/
  /* PERFORM MATH AND PRINT IT OUT*/
  CALL PRINTSTRING (.HEAD1+3,20);
  CALL PRINTSTRING (.SCHK, LENGTH(SCHK));
  CALL PRINTSTRING (.N,1);
  CALL CAL; CALL PRINTSREPORT;
  GO TO START;
EXIT:;
DECLARE PRNTEND DATA(CR,LF,'END ');
CALL PRINTSTRING (.PRNTEND, LENGTH(PRNTEND));
CALL PRINTSTRING (.SCHK, LENGTH(SCHK)); CALL CRFL;
END SOURCESCHECK;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

ALTSFACTOR: PROCEDURE;
    DO M = 1 TO 2;
    CALL SEL$HEAD;
    CALL PRINTSTRING(.ALTMSG, LENGTH(ALTMSG));
    CALL IMP$LINE(.ALT$COR$FACTOR);
    DO I = 0 TO CNT; ALT$COR$FACTOR(I) = ALT$COR$FACTOR(I) - 30H; END;
    ALT$COR$FACTOR(13) = 24H;
    IF SELDET = 0 THEN
        DO I = 0 TO 13; ALT$FACTOR1(I) = ALT$COR$FACTOR(I); END;
    ELSE
        DO I = 0 TO 13; ALT$FACTOR2(I) = ALT$COR$FACTOR(I); END;
    END;
    END ALTSFACTOR;

/* MAIN */

DECLARE SIGNON DATA('REMOTE WORKING LEVEL MONITOR VERSION', '11 04-26-77', CR, LF);

RESET:
    OUTPUT(RNL$HEAD) = FALSE; /*RESET READ*/
    DO I = 0 TO 5; YRS(I) = 0; END; /*RESET DATE AND TIME TO 0*/
    CAL$TEST = FALSE; /*BE SURE NOT IN CAL MODE*/
    CAL$TEST1 = FALSE; CAL$TEST2 = FALSE;
    /*SET ALTITUDE CORRECTION FACTOR TO 1.0*/
    DO I = 0 TO 12; ALT$FACTOR1(I) = 0; ALT$FACTOR2(I) = 0; END;
    ALT$FACTOR1(0) = 1; ALT$FACTOR1(1) = 24H;
    ALT$FACTOR2(0) = 1; ALT$FACTOR2(1) = 24H;

CNT = 1; SELDET = 0;
START: CALL CRLF;
    CALL PRINTSTRING(.SIGNON, LENGTH(SIGNON));
    CALL TIPE$DATE;
    CALL TIPE$PROMPT;
    DO I = 0 TO 7; BIT$P(07:I) = 0; BIT$P(04:I) = 0; END; /*CLEAR CONTROL VECTOR*/
    CALL TIPE$PROMPT;
    /*INPUT CHAR FROM TTY, ECHO, Branch*/
    OUTPUT(OFBH) = 0; ENABLE;
    CALL TIPE$OUT;
    /*PROCESS TIPE$ COMMAND INTO NUMEEP AND EXECUTE*/
    IF (W = (W - 40H)) > 19
        THEN GOTO MESSAGE;
REMOTE WORKING LEVEL MONITOR CONTROL PROGRAM

/* THIS SECTION SELECTS THE BRANCH SELECTED BY THE KBED*/

DO CASE N;
  GOTO MESSAGE; /* CASE 0 */
  CALL ALT$FACTOR; /* CHAR A */
  GOTO MESSAGE; /* CHAR = B */
  CALL CALIBRATE; /* CHAR = C */
  CALL CHANGEDATE; /* CHAR = D */
  CALL CAL; /* CHAR = E */
  CALL TYPEDATE; /* CHAR = F */
  CALL MAIN; /* CHAR = G */
  CALL SEL$HEAD; /* CHAR = H */
  GOTO MESSAGE; /* CHAR = I */
  CALL PAPADV; /* CHAR = J */
  GO TO 3800H; /* CHAR = K, RETURN TO MONITOR */
  GOTO MESSAGE; /* CHAR = L */
  GOTO MESSAGE; /* CHAR = M */
  CALL PRINTREPORT; /* CHAR = P */
  DC:
  CALL READ$ACCUM(.B0,.B1,.B2); CALL READ$ACCUM(.D0,.D1,.D2):
  END; /* CHAR = Q */
  GOTO RESET; /* CHAR = R */
  CALL SOUR$CHECK; /* CHAR = S */
  END;
  GOTO LOOP;
MESSAGE: CALL PRINTSTRING(.LST,LENGTH(LST));
GOTO LOOP;
EOF;
| C00000020 | C00000006 | C00000007 | C00000008 | C00000009 | C00000010 | C00000011 | C00000012 | C00000013 | C00000014 | C00000015 | C00000016 | C00000017 | C00000018 | C00000019 | C00000020 | C00000021 | C00000022 | C00000023 | C00000024 | C00000025 | C00000026 | C00000027 | C00000028 | C00000029 | C00000030 | C00000031 | C00000032 | C00000033 | C00000034 | C00000035 | C00000036 | C00000037 | C00000038 | C00000039 | C00000040 | C00000041 | C00000042 | C00000043 | C00000044 | C00000045 | C00000046 | C00000047 | C00000048 | C00000049 | C00000050 | C00000051 | C00000052 | C00000053 | C00000054 | C00000055 | C00000056 | C00000057 | C00000058 | C00000059 | C00000060 | C00000061 | C00000062 |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
PLM TO ASSEMBLY LANGUAGE CROSS INDEX

<table>
<thead>
<tr>
<th>Memory Location</th>
<th>Assembly Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000063</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000064</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000065</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000066</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000067</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000068</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000069</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000070</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000071</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000072</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000073</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000074</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000075</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000076</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000081</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000082</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000083</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000084</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000085</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000086</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000087</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000088</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000089</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000090</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000091</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000092</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000093</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000094</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000095</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000096</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000097</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000098</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000099</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000100</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000101</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000102</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000103</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000104</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000105</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000106</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000107</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000108</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000109</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000110</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000111</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000112</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000113</td>
<td>SYSIN</td>
</tr>
<tr>
<td>00000114</td>
<td>SYSIN</td>
</tr>
</tbody>
</table>

STACK SIZE = 36 BYTES
<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000117</td>
<td>SYSIN</td>
<td>2200H</td>
</tr>
<tr>
<td>00000118</td>
<td>SYSIN</td>
<td>2010H</td>
</tr>
<tr>
<td>00003119</td>
<td>CONSTR</td>
<td>0110H</td>
</tr>
<tr>
<td>0000120</td>
<td>CTT</td>
<td>2049H</td>
</tr>
<tr>
<td>0000121</td>
<td>EBIT07</td>
<td>2049H</td>
</tr>
<tr>
<td>0000122</td>
<td>EBIT04</td>
<td>2052H</td>
</tr>
<tr>
<td>0000123</td>
<td>EBIT1</td>
<td>2054H</td>
</tr>
<tr>
<td>0000124</td>
<td>EBIT2</td>
<td>2056H</td>
</tr>
<tr>
<td>0000125</td>
<td>EBIT4</td>
<td>2057H</td>
</tr>
<tr>
<td>0000126</td>
<td>EBIT5</td>
<td>2067H</td>
</tr>
<tr>
<td>0000127</td>
<td>EBIT6</td>
<td>2068H</td>
</tr>
<tr>
<td>0000128</td>
<td>EBIT7</td>
<td>2069H</td>
</tr>
<tr>
<td>0000131</td>
<td>EBIT8</td>
<td>2069H</td>
</tr>
<tr>
<td>0000132</td>
<td>EBIT9</td>
<td>2069H</td>
</tr>
<tr>
<td>0000133</td>
<td>EBIT10</td>
<td>2070H</td>
</tr>
<tr>
<td>0000134</td>
<td>EBIT11</td>
<td>2070H</td>
</tr>
<tr>
<td>0000135</td>
<td>EBIT12</td>
<td>2070H</td>
</tr>
<tr>
<td>0000136</td>
<td>EBIT13</td>
<td>2070H</td>
</tr>
<tr>
<td>0000137</td>
<td>EBIT14</td>
<td>2070H</td>
</tr>
<tr>
<td>0000138</td>
<td>EBIT15</td>
<td>2070H</td>
</tr>
<tr>
<td>0000139</td>
<td>EBIT16</td>
<td>2070H</td>
</tr>
<tr>
<td>0000140</td>
<td>EBIT17</td>
<td>2070H</td>
</tr>
<tr>
<td>0000141</td>
<td>EBIT18</td>
<td>2070H</td>
</tr>
<tr>
<td>0000142</td>
<td>EBIT19</td>
<td>2070H</td>
</tr>
<tr>
<td>0000143</td>
<td>EBIT20</td>
<td>2070H</td>
</tr>
<tr>
<td>0000144</td>
<td>EBIT21</td>
<td>2070H</td>
</tr>
<tr>
<td>0000145</td>
<td>EBIT22</td>
<td>2070H</td>
</tr>
<tr>
<td>0000146</td>
<td>EBIT23</td>
<td>2070H</td>
</tr>
<tr>
<td>0000147</td>
<td>EBIT24</td>
<td>2070H</td>
</tr>
<tr>
<td>0000148</td>
<td>EBIT25</td>
<td>2070H</td>
</tr>
<tr>
<td>0000149</td>
<td>EBIT26</td>
<td>2070H</td>
</tr>
<tr>
<td>0000150</td>
<td>EBIT27</td>
<td>2070H</td>
</tr>
<tr>
<td>0000151</td>
<td>EBIT28</td>
<td>2070H</td>
</tr>
<tr>
<td>0000152</td>
<td>EBIT29</td>
<td>2070H</td>
</tr>
<tr>
<td>0000153</td>
<td>EBIT30</td>
<td>2070H</td>
</tr>
<tr>
<td>0000154</td>
<td>EBIT31</td>
<td>2070H</td>
</tr>
<tr>
<td>0000155</td>
<td>EBIT32</td>
<td>2070H</td>
</tr>
<tr>
<td>0000156</td>
<td>EBIT33</td>
<td>2070H</td>
</tr>
<tr>
<td>0000157</td>
<td>EBIT34</td>
<td>2070H</td>
</tr>
<tr>
<td>0000158</td>
<td>EBIT35</td>
<td>2070H</td>
</tr>
<tr>
<td>0000159</td>
<td>EBIT36</td>
<td>2070H</td>
</tr>
<tr>
<td>0000160</td>
<td>EBIT37</td>
<td>2070H</td>
</tr>
<tr>
<td>0000161</td>
<td>EBIT38</td>
<td>2070H</td>
</tr>
<tr>
<td>0000162</td>
<td>EBIT39</td>
<td>2070H</td>
</tr>
<tr>
<td>0000163</td>
<td>EBIT40</td>
<td>2070H</td>
</tr>
<tr>
<td>0000164</td>
<td>EBIT41</td>
<td>2070H</td>
</tr>
<tr>
<td>0000165</td>
<td>EBIT42</td>
<td>2070H</td>
</tr>
<tr>
<td>0000166</td>
<td>EBIT43</td>
<td>2070H</td>
</tr>
<tr>
<td>0000167</td>
<td>EBIT44</td>
<td>2070H</td>
</tr>
<tr>
<td>0000168</td>
<td>EBIT45</td>
<td>2070H</td>
</tr>
<tr>
<td>0000169</td>
<td>EBIT46</td>
<td>2070H</td>
</tr>
<tr>
<td>0000170</td>
<td>EBIT47</td>
<td>2070H</td>
</tr>
<tr>
<td>0000171</td>
<td>EBIT48</td>
<td>2070H</td>
</tr>
<tr>
<td>0000172</td>
<td>EBIT49</td>
<td>2070H</td>
</tr>
<tr>
<td>0000173</td>
<td>EBIT50</td>
<td>2070H</td>
</tr>
<tr>
<td>0000174</td>
<td>EBIT51</td>
<td>2070H</td>
</tr>
<tr>
<td>Variable</td>
<td>Location in RAM</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>DS</td>
<td>211CH</td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>211DB</td>
<td></td>
</tr>
<tr>
<td>CALTST</td>
<td>211FD</td>
<td></td>
</tr>
<tr>
<td>CALTST2</td>
<td>2120H</td>
<td></td>
</tr>
<tr>
<td>CALTST3</td>
<td>2121H</td>
<td></td>
</tr>
<tr>
<td>AIIFACTOR1</td>
<td>2123H</td>
<td></td>
</tr>
<tr>
<td>AIIFACTOR2</td>
<td>2130H</td>
<td></td>
</tr>
<tr>
<td>TPIXOUT</td>
<td>033CB</td>
<td></td>
</tr>
<tr>
<td>CHAR</td>
<td>213DB</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>213FH</td>
<td></td>
</tr>
<tr>
<td>TYPFIN</td>
<td>0373B</td>
<td></td>
</tr>
<tr>
<td>TYPINOUT</td>
<td>0376H</td>
<td></td>
</tr>
<tr>
<td>FCBO</td>
<td>0380H</td>
<td></td>
</tr>
<tr>
<td>CFP</td>
<td>0393F</td>
<td></td>
</tr>
<tr>
<td>IMPLINE</td>
<td>0395H</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>2140H</td>
<td></td>
</tr>
<tr>
<td>CFP</td>
<td>0340H</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>0349H</td>
<td></td>
</tr>
<tr>
<td>FFPOR</td>
<td>0354H</td>
<td></td>
</tr>
<tr>
<td>FPRMSG</td>
<td>0367H</td>
<td></td>
</tr>
<tr>
<td>STAPT</td>
<td>036FH</td>
<td></td>
</tr>
<tr>
<td>RELAYMS</td>
<td>037FH</td>
<td></td>
</tr>
<tr>
<td>RELAYSHS</td>
<td>0384H</td>
<td></td>
</tr>
<tr>
<td>TYPEDATE</td>
<td>0395H</td>
<td></td>
</tr>
<tr>
<td>TAMP</td>
<td>040FH</td>
<td></td>
</tr>
<tr>
<td>TINP</td>
<td>0416H</td>
<td></td>
</tr>
<tr>
<td>SETYEAR</td>
<td>0429H</td>
<td></td>
</tr>
<tr>
<td>SETH</td>
<td>0476H</td>
<td></td>
</tr>
<tr>
<td>SETD</td>
<td>0477H</td>
<td></td>
</tr>
<tr>
<td>SETP</td>
<td>0478H</td>
<td></td>
</tr>
<tr>
<td>SETH</td>
<td>0493H</td>
<td></td>
</tr>
<tr>
<td>CHANGEDATE</td>
<td>0508H</td>
<td></td>
</tr>
<tr>
<td>PSTART</td>
<td>0519H</td>
<td></td>
</tr>
<tr>
<td>TKEEPER</td>
<td>0527H</td>
<td></td>
</tr>
<tr>
<td>DAYMAX</td>
<td>0542H</td>
<td></td>
</tr>
<tr>
<td>RETS5G</td>
<td>0546H</td>
<td></td>
</tr>
<tr>
<td>COM5G</td>
<td>0557H</td>
<td></td>
</tr>
<tr>
<td>DETHPADOUT</td>
<td>0575H</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>2148H</td>
<td></td>
</tr>
<tr>
<td>CMOUT</td>
<td>0620H</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>2147H</td>
<td></td>
</tr>
<tr>
<td>PLEASERELEASE</td>
<td>0626H</td>
<td></td>
</tr>
<tr>
<td>SGN5G</td>
<td>0629H</td>
<td></td>
</tr>
<tr>
<td>PAPADY</td>
<td>0746H</td>
<td></td>
</tr>
<tr>
<td>PULSES</td>
<td>2169H</td>
<td></td>
</tr>
<tr>
<td>FINISH</td>
<td>069AH</td>
<td></td>
</tr>
<tr>
<td>WOPAPER</td>
<td>0737H</td>
<td></td>
</tr>
<tr>
<td>FX2ADVANCE</td>
<td>0755H</td>
<td></td>
</tr>
<tr>
<td>Variable Location in RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000233 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000234 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000235 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000236 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000237 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000238 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000239 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000240 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000241 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000242 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000243 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000244 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000245 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000246 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000247 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000248 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000249 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000250 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000251 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000252 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000253 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000254 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000255 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000256 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000257 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000258 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000259 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000260 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000261 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000262 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000263 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000264 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000265 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000266 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000267 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000268 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000269 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000270 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000271 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000272 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000273 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000274 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000275 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000276 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000277 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000278 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000279 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000280 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000281 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000282 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000283 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000284 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000285 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000286 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000287 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000288 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000289 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'0000290 SYSIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIABLE LOCATION IN PAN</td>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>LIST......................................</td>
<td>1505H</td>
<td></td>
</tr>
<tr>
<td>SELECTHEAD...............................</td>
<td>1524H</td>
<td></td>
</tr>
<tr>
<td>SYSLHEAD..................................</td>
<td>1539H</td>
<td></td>
</tr>
<tr>
<td>FXT......................................</td>
<td>1592H</td>
<td></td>
</tr>
<tr>
<td>FINISH....................................</td>
<td>1593H</td>
<td></td>
</tr>
<tr>
<td>LOOP.......................................</td>
<td>1574H</td>
<td></td>
</tr>
<tr>
<td>FSTAPTACCUM..............................</td>
<td>1591H</td>
<td></td>
</tr>
<tr>
<td>PRINTCALDATA..............................</td>
<td>1599H</td>
<td></td>
</tr>
<tr>
<td>CALHEAD.................................</td>
<td>15EDR</td>
<td></td>
</tr>
<tr>
<td>NETBGK....................................</td>
<td>15P9H</td>
<td></td>
</tr>
<tr>
<td>MLP1......................................</td>
<td>15P3M</td>
<td></td>
</tr>
<tr>
<td>MLP2......................................</td>
<td>15F9H</td>
<td></td>
</tr>
<tr>
<td>MLP3......................................</td>
<td>15P9F</td>
<td></td>
</tr>
<tr>
<td>MLP4......................................</td>
<td>1605H</td>
<td></td>
</tr>
<tr>
<td>CALSUB.................................</td>
<td>1634H</td>
<td></td>
</tr>
<tr>
<td>CALBRATE...............................</td>
<td>16A3H</td>
<td></td>
</tr>
<tr>
<td>F1........................................</td>
<td>21F4H</td>
<td></td>
</tr>
<tr>
<td>N2........................................</td>
<td>21F7H</td>
<td></td>
</tr>
<tr>
<td>FINISH.................................</td>
<td>176CH</td>
<td></td>
</tr>
<tr>
<td>SOURCECHECK.........................</td>
<td>175CH</td>
<td></td>
</tr>
<tr>
<td>SCHF....................................</td>
<td>1770H</td>
<td></td>
</tr>
<tr>
<td>SPLSOURCE..............................</td>
<td>1783H</td>
<td></td>
</tr>
<tr>
<td>I.........................................</td>
<td>21F7H</td>
<td></td>
</tr>
<tr>
<td>START.................................</td>
<td>1799H</td>
<td></td>
</tr>
<tr>
<td>FXT.................................</td>
<td>1911H</td>
<td></td>
</tr>
<tr>
<td>FNETEND...............................</td>
<td>1912H</td>
<td></td>
</tr>
<tr>
<td>ALTFACTOR..............................</td>
<td>1938H</td>
<td></td>
</tr>
<tr>
<td>SIGMON.................................</td>
<td>19P7H</td>
<td></td>
</tr>
<tr>
<td>PESFT.................................</td>
<td>124H</td>
<td></td>
</tr>
<tr>
<td>MESSAGE...............................</td>
<td>1F9PH</td>
<td></td>
</tr>
</tbody>
</table>
PLM COMPILER ASSEMBLY LANGUAGE OUTPUT

0000383 SYSTN
0000384 SYSTN
0000385 SYSTN
0000386 SYSTN
0000387 SYSTN
0000388 SYSTN
0000389 SYSTN
0000390 SYSTN
0000391 SYSTN
0000392 SYSTN
0000393 SYSTN
0000394 SYSTN
0000395 SYSTN
0000396 SYSTN
0000397 SYSTN
0000398 SYSTN
0000399 SYSTN
0000400 SYSTN
0000401 SYSTN
0000402 SYSTN
0000403 SYSTN
0000404 SYSTN
0000405 SYSTN
0000406 SYSTN
0000407 SYSTN
0000408 SYSTN
0000409 SYSTN
0000410 SYSTN
0000411 SYSTN
0000412 SYSTN
0000413 SYSTN
0000414 SYSTN
0000415 SYSTN
0000416 SYSTN
0000417 SYSTN
0000418 SYSTN
0000419 SYSTN
0000420 SYSTN
0000421 SYSTN
0000422 SYSTN
0000423 SYSTN
0000424 SYSTN
0000425 SYSTN
0000426 SYSTN
0000427 SYSTN
0000428 SYSTN
0000429 SYSTN
0000430 SYSTN
0000431 SYSTN
0000432 SYSTN
0000433 SYSTN
0000434 SYSTN
0000435 SYSTN
0000436 SYSTN
0000437 SYSTN
0000438 SYSTN
0000439 SYSTN
0000440 SYSTN
PIN COMPILER ASSEMBLY LANGUAGE OUTPUT

00008887 SYSIN
00008488 SYSIN
1574H 0DH 0AH 4EH 4FH 54H 20H 50H 3FH 53H 4FH 42H 4CH 45H
00008489 SYSIN
1582H LXI B 74H 15H MOV EI 0EH CALL 0EH 09H 02H LIX IP 03H 0CH
00008500 SYSIN
1594H LXI H 52H 20H DAD B 0FH 07H LXY 0FH 07H LIX H 00H 08H
00008501 SYSIN
159DR 06H 01H DAD B MOV AP POP H MOV MA LIX H 00H 08H 04H 02H LIX H
00008502 SYSIN
1596H LXI H 32H 20H DAD R PUSH H LXY 0FH 04H 02H LIX H
00008503 SYSIN
15AFH 06H 01H DAD B MOV AM PCP H MOV MA CALL APH 0FH 06H
00008504 SYSIN
15B8H INH 15H 1NH
00008505 SYSIN
15BCH 20H 4DH 49H 4FH 4FH 45H 20H 43H AFR 5FH 0FH 0DH 00H 01H 05H
00008506 SYSIN
15C9H LXI B 03H 00H LIX H 09H 0CH DAD B XCRG MOV CP
00008507 SYSIN
15D5H MOV BP MOV EI 14H CALL 09H 04H MOV LI 22H MOV CP
00008508 SYSIN
15DEH CALL 34H 04H LIX B RXC 15H MOV Pt 10H CALL
00008509 SYSIN
15FH 09H 04H CALL 69H 0CH RFP CALL 9CH 06H
00008510 SYSIN
15F0H 01H 01H 16H
00008511 SYSIN
15F3H 02H FFP 05H 00H 00H 24H
00008512 SYSIN
15F9H 01H 05H FEH 00H 00H 24H
00008513 SYSIN
15FFH 01H 07H FEH 00H 00H 24H
00008514 SYSIN
1605H 01H FEH 00H 00H 24H
00008515 SYSIN
1608H DCB H MOV LI CEH MOV MI OCH MOV AI 02H LIX H 0F
00008516 SYSIN
1614H 20H SUB H JC 00H 1NH MOV CP 0CH MOV DI 0CH
00008517 SYSIN
161DH LXI H D0H 20H MOV CH *MOV BI CIX CALL FCH CP
00008518 SYSIN
1626H LXI B F3H 15H XCRG DAD B XCRG MOV CE MOV BP CALL
00008519 SYSIN
162FH 09H 08H LXI B A9H 21H LXY * 21H 20H MOV MC
0000851A SYSIN
1639H INR L MOV MB MOV EX 04H MOV DI 00H LIX H CEP 2FH
0000851B SYSIN
1641H MOV CM MOV BI 00H CALL FCH CPN LIX H 20H MOV CP
0000851C SYSIN
164AH MOV CM MOV BM XCRG DAD B XCRG MOV CE MOV BP CALL
0000851D SYSIN
1653H 81H 08H LXI B A9H 21H LIX H 21H 20H MOV MC
0000851E SYSIN
165CH INR L MOV MB MOV EX 04H CALL FCH CPN LIX H CEP 2FH
0000851F SYSIN
1665H MOV CM MOV BI 00H CALL FCH CPN LIX H 21H MOV CP
00008520 SYSIN
1666H MOV CM INR L MOV BM XCRG DAD B XCRG MOV CE MOV BP CALL
00008521 SYSIN
1679H 55H 08H LXI H CEP 21H INR P 10H JNZ LEX 10H 16H
00008522 SYSIN
1680H CALL DOH 00H CALL 00H RET LIX P 9FH
00008523 SYSIN
168B9H 20H INR H MOV LI F0H MOV MC INS MOV MB LIX P A1H
00008524 SYSIN
1692H 20H LXI D A9H 20H CALL ADH 10H CALL 01H
00008525 SYSIN
1699H 15H CALL EDH 15H CALL FDA 15H RFT LIX P
00008526 SYSIN
16A4H FDH 21H MOV MI 01H MOV AI 02H LIX P FDH 21H
00008527 SYSIN
16A8H SUB H JC 00H 1NH MOV CP 0CH MOV CM DCP C 1NH 0CH 16H
00008528 SYSIN
16B6H DCB H MOV LI B2H MOV MI 0CH MOV LI 4AH MOV MI 00H
00008529 SYSIN
16BH JMP CEP 16H DCR H MOV LI P2H MOV MI 01H MOV LI
0000852A SYSIN
16CBH 4AH MOV MI FFH INR H MOV LI 0FH MOV MI 00H MOV LI
0000852B SYSIN
16DFH 09H LXI H FFH 21H SUB H JC 67H 17H YPA A
0000852C SYSIN
16D9H OUT BHI F6H MOV WI 01H LIX H FFH INR L MOV MI FFH JNP L
0000852D SYSIN
16E3H 12H CALL 91H 15H CALL 0CH 00H CALL 6FH
0000852E SYSIN
16EC8H 00H MOV LI 20H MOV MI FFH INS MOV MI FFH JNP L
0000852F SYSIN
16F5H MOV MI OSH DCB H MOV LI 0DH MOV MI 2CH INT H MOV LI
00008530 SYSIN
16F7H 02H MOV LI 00H MOV MI OCH CALL 84H 16H MOV LI
00008531 SYSIN
16F8H 22H MOV LI 30H DCR H MOV LI 0AH MOV MI 0AH MOV LI 16H
00008532 SYSIN
1707H 15H MOV MI 00H MOV LI 00H MOV LI 00H MOV MI 3H
00008533 SYSIN
1722H INR H MOV MI 09H MOV LI 00H MOV MI 02H MOV LI
00008534 SYSIN
1726H 15H DCR H MOV LI DOH MOV MI 03H INR H MOV LI 22H
00008535 SYSIN
1739H MOV MI 39H MOV LI 20H MOV MI 0AH DCR L MOV MI 00H
00008536 SYSIN
173DH 15H MOV LI 24H MOV MI 14H INT H MOV MI 0AH INT L
00008537 SYSIN
1746H MOV MI 6CH INR H MOV MI 14H MOV MI 16H
00008538 SYSIN
174FH 21H MOV MI 00H INH 0EH MOV CA MOV MI 17H SUP L
00008539 SYSIN
1755H 08H SUB I 01H SBC A A9A C PPC JC 6CH 17H
0000853A SYSIN
1761H MOV LI FEH INR H JNZ DCH 16H DCR I INR H JNZ
0000853B SYSIN
176AH 08H 16H NET JMP 99H 17H
PIN COMPILER ASSEMBLY LANGUAGE OUTPUT

00001021 SYSIN
00001022 SYSIN
00001023 SYSIN
00001024 SYSIN
00001025 SYSIN

1B848 13H 188 19H 18H 1CH 1BH 22H 1BH 25H 19H 1AH 18H 2EH 1BH 31H 1BH
1B948 34H 18H 3AH 1BH 61H 1BH 6AH 1BH
1BP9C JMP C9H 1AH LIX B 05H 15H MOV FI 1FH CALI
1B458 09H 06H JMP C9H 1AH FI RT*
NO PROGRAM ERRORS
1 REM RWLM CALIBRATION PROGRAM
2 REM ARGONNE NATIONAL LABORATORY ELECTRONICS DIVISION
20 PRINT "RWLMCAL.RAS V02.1": GOSUB 4000
30 FOR I=0 TO 9
35 READ C(I),CO(I)
40 NEXT I
50 DATA .926838,1.918670,-3.196779,0.2,506266
51 DATA -.979403,1.701921,-11.12606,-18.620853,2.752940,16.756924
52 DATA .049957,-3.6592397,4.232080,12.896776,-.2515410-10.308247
100 S5=A5+C5:S3=A3+C3:S2=A2+C2
110 T1=A5-VO*C5:T2=(1+VO)*C5:T3=C1+VO)*C3:T4=A-VO*C
120 T5=(1+VO)*C
130 FOR 0=0 TO 6 STEP 3
140 F1(Q/3)=CC(O)*TI+CC0+1)*T2+C(Q+2)*T3)/cE1*V)
150 F2(Q/3)=(CO(0)*S5+COCQ+1)*S3+C(0Q+2)*S2)/CE1*V)
160 NEXT 0
170 W1=C13.68*F1(0)+7.68*(F(1)+F1(2)))/130000
180 W2=(13.68*F2C0)+7.68*CF2(1)+F2C2)))/130000
190 W3=1/K/E1*2*V*150)
200 IF S0=1 THEN GOSUB 2000:GOTO 250
210 INPUT "ER,EC PLEASE";E2,E3
250 GOSUB 3000
260 P1=.227261*FO01)/2.22:P2=.025864*FOC2)/2.22:P3=.035185*FOc3)/2.22
270 P4=(.13.68*FO(1)+.23.68*(FO(2)+FO(3)))/130000
280 F9=W(1)*A+W(2)*R3+WC3)*C
299 PRINT:PRINT:PRINT:PRINT
300 PRINT "R W L M C A L I R A T I O N":PRINT
301 PRINT:PRINT "FNA=";F1(0),,"FNR=";F1(1),,'FNC="F1
305 PRINT:PRINT "FNA2=";F2(0),"FNB2="iF2(1),"FNC2="F2(2)
310 PRINT:PRINT "WL1="1W1,"WLKUS="3W3,"ER="3E2,"EC="E3
312 PRINT:PRINT "RN-DAUGHTER COEFFICIENTS FOR IWL-MEMORY"
320 FOR I=1 TO 3: FØR J=1 TO 3
335 PRINT P1,P2,P3
340 PRINT:PRINT"RN-DAUGHTER (RWLM) IN ATOMS"
345 PRINT F0(1),F0(2),F0(3)
350 PRINT:PRINT"RN-DAUGHTER (RWLM) IN PCI/L"
355 PRINT P4,F9
360 PRINT:PRINT"WL FROM IWL MEM (DIRECT AND FROM RN-DAUGHTERS)"
365 PRINT P4,F9
370 PRINT W(1),W(2),W(3)
375 PRINT:PRINT:PRINT:PRINT:PRINT
380 PRINT"NORMAL STOP AT LINE 500":STOP
A3(2) = 0.006941 * E1 * V; A3(3) = 0.131 * E1 * V

C1(1) = (A2(2) * A3(3) - A3(2) * A2(3)) / B4
C2(1) = (A2(3) * A3(1) - A3(2) * A2(1)) / B4
C2(2) = (A1(1) * A3(3)) / B4
C2(3) = -(A1(1) * A2(3)) / B4
C3(1) = (A2(1) * A3(2) - A3(2) * A2(1)) / B4
C3(2) = -(A1(1) * A3(2)) / B4
C3(3) = (A1(1) * A2(2)) / B4

P(1,1) = C1(1) * 0.227261 / 2.22
P(1,2) = 0
P(1,3) = -P(1,1) * V0; F0(1) = C1(1) * T4
F0(2) = C2(1) * T4 + C2(2) * R3 + C2(3) * T5
P(2,1) = C2(1) * 0.025864 / 2.22; P(2,2) = C2(2) * 0.025864 / 2.22
P(2,3) = (C2(3) * (1 + V0) - C2(1) * V0) * 0.025864 / 2.22
F0(3) = C3(1) * T4 + C3(2) * B3 + C3(3) * T5
P(3,1) = C3(1) * 0.035185 / 2.22; P(3,2) = C3(2) * 0.035185 / 2.22
P(3,3) = (C3(3) * (1 + V0) - C3(1) * V0) * 0.035185 / 2.22
W(1) = (13.68 * C1(1) + 7.68 * (C2(1) + C3(1))) / 130000
W(2) = 7.68 * C2(2) + C3(2) / 130000
W(3) = (7.68 * (1 + V0) * (C2(3) + C3(3))) / 130000 - V0 * W(1)
RETURN

INPUT "CALCULATE OR INPUT EB, EC (C/I)"; R$ AS = "DATE, PLACE"
IF NOT R$ = "I" OR R$ = "C" THEN 4000
IF R$ = "I" THEN S0 = 0
IF R$ = "C" THEN S0 = 1
INPUT "DATE, PLACE"; D$, P$
INPUT "FLOW RATE (LITERS/MIN)"; V
INPUT "EFFICIENCY OF ALPHA DETECTOR"; E1
INPUT "OVLAP"; VO
INPUT "TOTAL BETA COUNTS FROM RAB AND RAC DURING 5 MINUTES"; Q1
INPUT "TOTAL BETA COUNTS FROM RAB AND RAC DURING 30 MINUTES"; Q2
INPUT "TOTAL ALPHA COUNTS IN RAA CHANNEL DURING 30 MINUTES"; A3
INPUT "TOTAL ALPHA COUNTS IN RAA CHANNEL DURING 35 MINUTES"; A2
INPUT "TOTAL ALPHA COUNTS IN RAC' CHANNEL DURING 35 MINUTES"; C2
INPUT "TOTAL ALPHA COUNTS IN RAC' CHANNEL DURING 5 MINUTES"; C5
INPUT "TOTAL ALPHA COUNTS IN RAC' CHANNEL DURING 2 MINUTES"; A5
INPUT "TOTAL BETA COUNTS FROM RAB AND RAC DURING 2 MINUTES"; R3
INPUT "TOTAL ALPHA COUNTS IN RAC' CHANNEL DURING 2 MINUTES"; C
INPUT "TOTAL ALPHA COUNTS FROM 39-41 MINUTES"; K1
RETURN
5000 END
0K
TABLE OF DEFINITIONS OF SYMBOLS IN CALIBRATION PROGRAM

A. Constants and Data Inputs
   \( V \) = Flowrate (liters/min)
   \( V_0 \) = Overlap of RaC\(^{'}\) counts
   \( A \) = RaA counts (2 min)
   \( A_5 \) = RaA counts (5 min)
   \( A_3 \) = RaA counts (30 min)
   \( A_2 \) = RaA counts (35 min)
   \( B_3 \) = Ra(B+C) counts (2 min)
   \( Q_1 \) = Ra(B+C) counts (5 min)
   \( Q_2 \) = Ra(B+C) counts (30 min)
   \( C \) = RaC\(^{'}\) count (2 min)
   \( C_5 \) = RaC\(^{'}\) count (5 min)
   \( C_3 \) = RaC\(^{'}\) count (30 min)
   \( C_2 \) = RaC\(^{'}\) count (35 min)
   \( E_1 \) = Alpha efficiency (EA)
   \( K_1 \) = 2-min Kusnetz count from (39 to 41 min)

B. Calculated Variables
   \( E_2 \) = RaB beta efficiency
   \( E_3 \) = RaC beta efficiency
   \[ F_{1(0)} = \text{FNA atoms/liter of RaA} \]
   \[ F_{1(1)} = \text{FNB atoms/liter of RaB} \]
   \[ F_{1(2)} = \text{FNC atoms/liter of RaC} \]
   \[ F_{2(0)} = \text{FNA2 atoms/liter of RaA} \]
   \[ F_{2(1)} = \text{FNB2 atoms/liter of RaB} \]
   \[ F_{2(2)} = \text{FNC2 atoms/liter of RaC} \]
   \( W_1 \) = WL\(_1\) (WL calculated from Rn-daughters derived by spectroscopic method)
   \( W_2 \) = WL\(_2\) (WL calculated from Rn-daughters derived by total-alpha method)
   \( W_3 \) = WLKUS (WL calculated by the Kusnetz method)

*Calculated by alpha-spectroscopic method.
**Calculated by total-alpha method.
\[ P(1,1), P(1,2), P(1,3) \]
\[ P(2,1), P(2,2), P(2,3) \quad \text{Rn-daughter coefficients} \]
\[ P(3,1), P(3,2), P(3,3) \quad \text{for RWLM memory RaA,} \]
\[ \text{RaB, RaC coefficients} \]

\[ F0(1), F0(2), F0(3) \quad \text{Rn-daughters (A, B, C) in} \]
\[ \text{atoms/liter calculated} \]
\[ \text{from previous coefficient} \]

\[ P1, P2, P3 \quad \text{Rn-daughters (A, B, C) in} \]
\[ \text{pCi/liter calculated from} \]
\[ \text{previous coefficient} \]

\[ F9 \quad \text{WL calculated directly} \]
\[ \text{from derived WL} \]
\[ \text{coefficients} \]

\[ F4 \quad \text{WL calculated from the Rn-} \]
\[ \text{daughters, F0's} \]

\[ W(1), W(2), W(3) \quad \text{WL coefficients for} \]
\[ \text{RWLM memory} \]
PROGRAM TO CAL PHYSICAL CONSTANTS FOR RWLM

R.T.WITEK/M.K.MCGEE 4-FEB-77

DOUBLE PRECISION AS(3), DELAY, T, TD2(4), ANSWER(4, 3, 3), B(9)
DOUBLE PRECISION D0(9), DA, DB, DC, DH, DJ, DG, DF, DE
DOUBLE PRECISION FLAM(3), AO(3), CS(3), AI(3), AL(3), A(3), AL(3), D(9)
TYPE 50
ACCEPT 2, T, TD2(1)
AO(1) = 0.0
AO(2) = 0.0
AO(3) = 0.0
FLAM(1) = 0.2272614
FLAM(2) = 0.0258637
FLAM(3) = 0.0351851
NUM = 3
N = 1
TD2(2) = 5
TD2(3) = 30
TD2(4) = 35
DO 70 II = 1, 4
DO 70 JJ = 1, 3
DO 90 I = 1, 3
90 C(I) = 0.0
C(JJ) = 1.0
TD = TD2(II)
CALL BTMNEQ(NUM, FLAM, AO, CS, T, AS, AI)
DO 100 I = 1, 3
100 CS(I) = 0.0
DELAY = 13.0/60.0
CALL BTMNEQ(NUM, FLAM, AS, CS, DELAY, AL, AI)
CALL BTMNEQ(3, FLAM, AL, CS, TD, AI)
DO 70 I = 1, 3
ANSWER(II, JJ, I) = AI(I)
70 CONTINUE
DO 6 II = 1, 4
DO 6 JJ = 1, 3
6 TYPE 7, (ANSWER(II, JJ, I), I = 1, 3)
K = 0
DO 10 JJ = 1, 3
10 K = K + 1
B(K) = ANSWER(2, JJ, 1)
CONTINUE
DO 11 II = 2, 3
DO 11 JJ = 1, 3
11 K = K + 1
B(K) = ANSWER(II, JJ, 3)
CONTINUE
C SPECTROSCOPIC DETERM & INVERTED MAT
TYPE 300, N
TYPE 8, (B(K), K = 1, 9)
D(1) = B(5) * B(9) - B(6) * B(8)
D(2) = B(3) * B(8) - B(2) * B(9)
D(3) = B(2) * B(6) - B(3) * B(5)
D(4) = B(6) * B(7) - B(4) * B(9)
D(5) = B(1) * B(9) - B(3) * B(7)
D(6) = B(3) * B(4) - B(1) * B(8)
D(7) = B(4) * B(8) - B(5) * B(7)
D(8) = B(2) * B(7) - B(1) * B(6)
D(9) = B(1) * B(5) - B(2) * B(4)
DA = B(1) * (B(5) * B(9) - B(6) * B(8))
DB = B(4) * (B(2) * B(9) - B(3) * B(8))
DC = B(7) * (B(2) * B(6) - B(5) * B(3))
DE=DA-DB+DC
DO 17 K=1,9
D(K)=D(K)/DE
CONTINUE
TYPE 400
N=N+1
K=K+1
III=1
DO 13 ,I=2,4
DO 13 JJ=1,3
K=K+1
IF (K.EQ.1) GO TO 55
IF (K.EQ.4) GO TO 55
IF (K.EQ.7) GO TO 55
B(K)=ANSWER(II,JJ,3)
CONTINUE
C TOTAL ALPHA DETERM & INVERTED MAT
N N+1
K=K-1
DO 20 II=3,2,-1
DO 20 JJ=1,3
K=K+1
CONTINUE
DO 30 II=2,3
DO 30 JJ=1,3
K=K+1
CONTINUE
GO TO 60
4 FORMAT(X,'X',II,= ',E16.10, ',E16.10)
3 FORMAT(1X,'X',II,= ',E16.10, ',E16.10)
15 FORMAT(IX,'Y',I1,=' ',X(E16.10,'.'),E16.10)
16 FORMAT(IX,'Z',I1,=' ',E16.10,' ',E16.10)
50 FORMAT(IX,' INPUT VARIABLES ')
200 FORMAT(/IX,F4.0,' MINUTE COUNTING TIME VALUES'/)
300 FORMAT(/IX, 'INVERSE MATRIX DATA FOR F',I1,' CONVERSION'/)
80 FORMAT(IX,(3(E28.10,2X))
400 FORMAT(/IX, 'SPECTROSCOPIC INVERTED MATRIX'/)
700 FORMAT(/IX, 'TOTAL ALPHA INVERTED MATRIX'/)
500 FORMAT(/IX, 'DAUGHTER COEFFICIENTS***2 MINUTE COUNTING TIME '/)
600 FORMAT(/IX, 'BETA COEFFICIENTS***5.30 MINUTE COUNTING TIME '/)
2 FORMAT(2F10.2)
7 FORMAT(IX,3(E16.10,2X))
55 III=III+1
B(K)-CANSWER(II,JJ,3))+(ANSWER(III,1,1))
GO TO 13
60 CALL PETER2(ANSWER,D0,D)
END

DOUBLE PRECISION ANSWER(4,3,3),D0(9),D(9)
DOUBLE PRECISION P(3,3),FM(3),WL(3)
100 TYPE 10
10 FORMAT(' INPUT PLEASE...')
ACCEPT 1, SWITCH
IF (SWITCH .EQ. 2.0) GOTO 900
1 FORMAT(F3.1)
ACCEPT 11,V,EA,OVLAP,BC5,BC30,A30,A35,C35
TYPE 200,V,EA,OVLAP,BC5,BC30,A30,A35,C35
200 FORMAT(1XF5.2,1X,F5.3,1X,F4.2,1X,5(F8.1,1X))
11 FORMAT(8F20.5)
12 TYPE 13
13 FORMAT(' COUNTS PLEASE...')
ACCEPT 14, A5,C5,C30,A,BC,C.CKUS
14 FORMAT(7F15.5)
210 FORMAT(IX,7(F10.2,1X))
S5=A5+C5
S30=A30+C30
S35=C35+A35
TA5 = A5 - OVLAP*C5
TC5 = (1. + OVLAP)*C5
TC30 = (1. + OVLAP)*C30
TA = A-OVLAP*C
TC = (1 + OVLAP) * C
FNA = (D(1)*TA5)/(EA*V)
FNB = (D(2)*TA5+D(5)*TC5+D(6)*TC30)/(EA*V)
FN2 = (D(2)*TA5+D(5)*TC5+D(6)*TC30)/(EA*V)
FNC = (D(7)*TA5+D(8)*TC5+D(9)*TC30)/(EA*V)
FN2C = (D(7)*TA5+D(8)*TC5+D(9)*TC30)/(EA*V)
WL1 = (13.68*FNA+7.68*(FNB+FNC))/130000.
WL2 = (13.68*FNB+7.68*(FNB+FNC))/130000.
WKUS = CKUS/(EA*2.*V*150.)
IF(SWITCH.EQ.1.0) GO TO 99
CALL BETEFF(V,FNA,FNB,FNC,EB,EC,ANSWER)
GO TO 88
99 TYPE 3
3 FORMAT( 'EB,EC PLEASE....' )
ACCEPT 2, EB,EC
2 FORMAT(2F10.5)
88 CALL MATINV(EA,EB,EC,V,OVLAP,TA,TC,BC,P,FM,WL,ANSWER)
.PA = .227261 * F(1)/2.22
```
P8 = .025864 * FM(2)/2.22
PC = .035185 * FM(3)/2.22
WUL = (13.68*FM(1) + 7.68*(FM(2)+FM(3)))/13000.
FIUL = WU(1)*A + WU(2)*BC + WU(3)*C
TYPE 15, FNA,FNB,FNC
TYPE 22, FNA2,FNB2,FNC2
22 FORMAT(' FNA2=',F8.1,' FNB2=',F8.1,' FNC2=',F8.1)
15 FORMAT(' FNA=',F6.1,' FNB=',F8.4,' FNC=',F8.1)
TYPE 1, UL1,WLKUS,EB,EC
16 FORMAT(' UL1=',F10.6,' WLKUS=',F8.6,' EB=',F8.4,' EC=',F8.4)
TYPE 21, UL2
21 FORMAT(' UL2=',F5.2)
TYPE 71
71 FORMAT(' RN-DAUGHTER COEFFICIENTS FOR IWLM-MEMORY

22 FORMAT(' RN-DAUGHTERS(IWLM) IN ATOMS AND PCI/LITER

18 FORMAT(D15.9,2X),3(F7.3,2X))
18 FORMAT(1X,3(D15.9,2X),3(F7.3,2X))

TYPE 72
2 FORMAT(' RN-DAUGHTERS(IWLM) IN ATOMS AND PCI/LITER')
18 FORMAT(D15.9,2X),3(F7.3,2X))
18 FORMAT(1X,3(D15.9,2X),3(F7.3,2X))

TYPE 73
3 FORMAT(' WL FRM 1 IWLM(DIRECT AND FROM RN-DAUGHTERS')
19 FORMAT(1X,2(F5.2,2X))
19 FORMAT(1X,2(F5.2,2X))

TYPE 74
4 FORMAT(' WL COEFFICIENTS FOR IWLM-MEMORY

20 FORMAT(1X,3(D20.10,2X))
20 FORMAT(1X,3(D20.10,2X))
GO TO 100
900 RETURN
END

MATRINV.FORT

SUBROUTINE MATINV(EA,EB,EC,V,OVLAP,TA,TC,BC,P,FML,)
DOUBLE PRECISION A(4,3,3),P(3,3),FM(3),WU(3)
A11=A(1,1,1)*EA*V
A12=.0
A13=.0
A21=A(1,1,2)*EB*V+A(1,1,3)*EC*V
A22=A(1,2,2)*EB*V+A(1,2,3)*EC*V
A23=A(1,3,3)*EC*V
A31=A(1,1,3)*EA*V
A32=A(1,2,3)*EA*V
A33=A(1,3,3)*EA*V
BD=A11*(A22*A33-A32*A23)
C11=(A22*A33-A23*A32)/BD
C21=(A23*A31-A21*A33)/BD
C22=(A11*A33)/BD
C23=(A11*A23)/BD
C31=(A21*A32-A22*A31)/BD
C32=(A11*A32)/BD
C33=(A11*A22)/BD
P(1,1)=C11*227261/2.22
P(1,2)=.0
P(1,3)=P(1,1)*OVLAP
FM(1)=C11*TA
FM(2)=C21*TA+C22*SC+C23*TC
P(2,1)=C21*.025864/2.22
P(2,2)=C22*.025864/2.22
P(2,3)=C23*(1.+OVLAP)-C21*OVLAP)*.025864/2.22
FM(3)=C31*TA+C32*SC+C33*TC
```
SUBROUTINE BTMNEQ
DOUBLE PRECISION FLAM(3), A0(3), C(3), ANK
DO i = 1, NUM
AICI = 0.0
CONTINUE
DO 40 N1 = 1, NUM
DO 30 K = N1, NUM
DO 20 N = N1, K
ANK = A0(N1)*FLAM(N)/FLAM(N1)-C(N1)
CONTINUE
DO 10 I = N1, K
IF (I-N) 9,10,9
ANK = ANK*FLAM(I)/(FLAM(I)-FLAM(N))
CONTINUE
A(K) = A(K) + ANK*(1.0 - DEXP(-FLAM(N)*T))/FLAM(N)
A(K) = A(K) + ANK*DEXP(-FLAM(N)*T)
A(K) = A(K) + C(N1) * T
CONTINUE
RETURN
END

P(3,1)=C31*.035185/2.22
P(3,2)=C32*.035185/2.22
P(3,3)=(C33*(1+OVLAP)-C31*OVLAP)*.035185/2.22
WL(2)=7.68*(C22+C32)/130000.
WL(3)=(7.68*(1.+OVLAP)*(C23+C33))/130000.-OVLAP*WL(1)
RETURN
END

BETEFF SUB

SUBROUTINE BETEFF(VFNA, FNBFNC5, BC30, EB, EC)
DOUBLE PRECISION A(4,3,3)
X1=(A(2,1,2)*FNA+A(2,2,2)*FNB)*V
X2=(A(3,1,3)*FNA+A(3,2,3)*FNB+A(3,3,3)*FNC)*V
X3=(A(2,1,2)*FNA+A(2,2,2)*FNB)*V
X4=(A(2,1,3)*FNA+A(2,2,3)*FNB)*V
DET = X1*X2-X3*X4
Y1 = (A(3,1,3)*FNA+A(3,2,3)*FNB+A(3,3,3)*FNC)*V*BC5
Y2 = (A(2,1,3)*FNA+A(2,2,3)*FNB)*V*BC30
DYTE= Y1-Y2
EB=DETY/DET
Z1=(A(2,1,2)*FNA+A(2,2,2)*FNB)*V*BC30
Z2=(A(3,1,3)*FNA+A(3,2,3)*FNB)*V*BC5
DETC=Z1-Z2
EC=DETC/DET
RETURN
END

BTMNEQ SUBROUTINE

SUBROUTINE BTMNEQ(NUM, FLAM, A0, C, T, A, AI)
DOUBLE PRECISION FLAM(3), A0(3), C(3), A(3), AI(3), T
DOUBLE PRECISION ANK
DO I = 1, NUM
AII(I) = 0.0
A(I) = 0.0
CONTINUE
DO 40 NI = 1, NUM
DO 30 K = NI, NUM
DO 20 N = NI, K
ANK = A0(NI)*FLAM(N)/FLAM(NI)-C(NI)
CONTINUE
DO 10 I = NI, K
IF (I-N) 9,10,9
ANK = ANK*FLAM(I)/(FLAM(I)-FLAM(N))
CONTINUE
A(K) = A(K) + ANK*(1.0 - DEXP(-FLAM(N)*T))/FLAM(N)
A(K) = A(K) + ANK*DEXP(-FLAM(N)*T)
A(K) = A(K) + C(N1) * T
CONTINUE
RETURN
END