(NASA-CR-161849) COAL FACE MEASUREMENT SYSTEM FOR UNDERGROUND USE Final Report (Benton Corp., Manor, Pa.) 99 p HC A05/MF A01 CSCL 08I

N81-32580

Unclas G3/43 27482

COAL FACE MEASUREMENT SYSTEM

FOR UNDERGROUND USE

FINAL REPORT

CONTRACT NUMBER NAS8-33792

COAL FACE MEASUREMENT SYSTEM

FOR UNDERGROUND USE

FINAL REPORT

Contract Number NAS8-33792

June 15, 1981

Prepared for:

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama 35812

Prepared by:

Benton Corporation
Race St.
Manor, PA 15665

412/863-1152

Report Number 262-400

INDEX

I N D E X (continued)

ILLUSTRATIONS

I L L U S T R A T I O N S (continued)

FORWARD

This report was prepared by the Benton Corporation of Manor, PA under contract NAS8-33792 issued by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Marshall Space Flight Center, AL. The work was performed under the supervision of the contracting officers' representatives, Messrs. James Currie and Ralph Kissel.

SUMMARY

This report presents the results of work performed in the development and testing of a Coal Face Measurement System. It begins with a review of the measurement method; and the techniques, hardware, and system operation procedures. The next portion of the report is devoted to a description of the tests performed at the Department of Energy facility at Bruceton, PA. Conclusions are given in the final section.

1.0 SYSTEM MEASUREMENT METHOD

1.1 Introduction

A measurement system was developed for the Eickhoff longwall shearer to determine the contour of the coal face as it mines coal. Contour data is obtained by an indirect measurement technique based on evaluating the motion of the shearer during mining. Starting from a known location, points along the coal face are established through a knowledge of the machines' positions and yaw movements as it moves past the coal face.

The equipment used in this system can be grouped into three parts. These are:

1) An angle transducer assembly
2) A distance transducer assembly
3) Data storage and reduction electronics

The angle transducers measure the angle between respective track sections as the shearer proceeds along the coal face. The distance transducer functions in conjunction with them to obtain relative angles at known positions. After completely cutting the coal face the accumulated data is stored on cassette tape and the present track profile is computed and displayed.

The transducer assemblies are shown in Figure 1.1. The data storage and reduction electronics is housed in an explosion proof enclosure. See Figure 1.1.1.

The coal face measuring equipment was constructed at the Benton Corporation and then transferred to the Department of Energy test facility at Bruceton, PA. Tests were performed there for three different coal face profiles by physically shaping the track. The system results were compared to manually measured values and after analysis shown to be within 00.1 foot of the correct contour.

1.2 Measurement Surface

The topography for a longwall mine complex is illustrated in Figure 1.2. The survey marks are points for which coordinates have been established. All other points between are unknown.

The technique used to measure the coal profile consists of measuring the angle between "Eicotrack" rack sections over which the shearer moves, and then performing trigonometric calculations to determine its relative shape. The rack sections are approximately two and one half feet in length and are assembled with conveyor "pan" sections to form a path of travel for the machine. The combined length

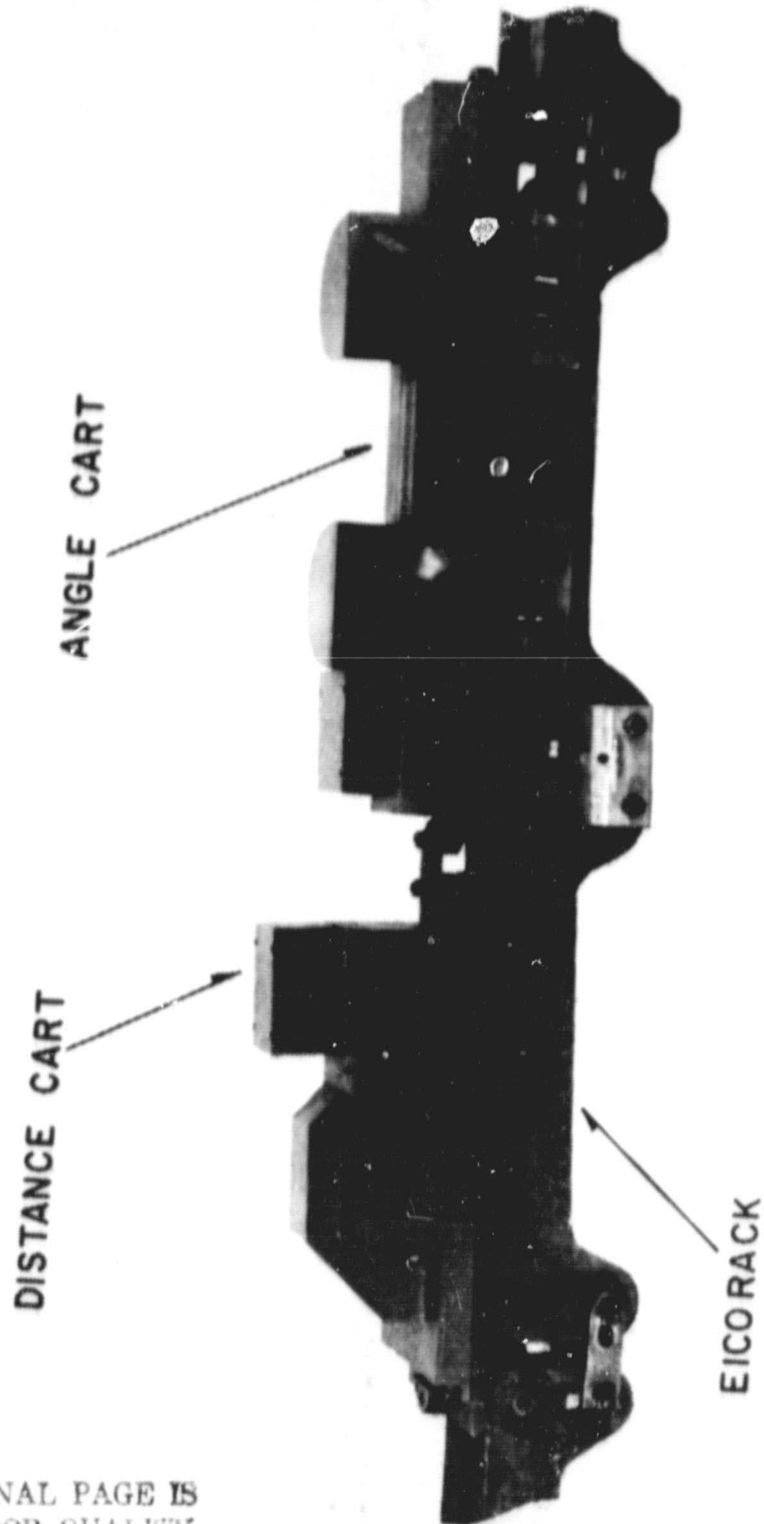


Figure 1.1 Angle and Distance Transducers

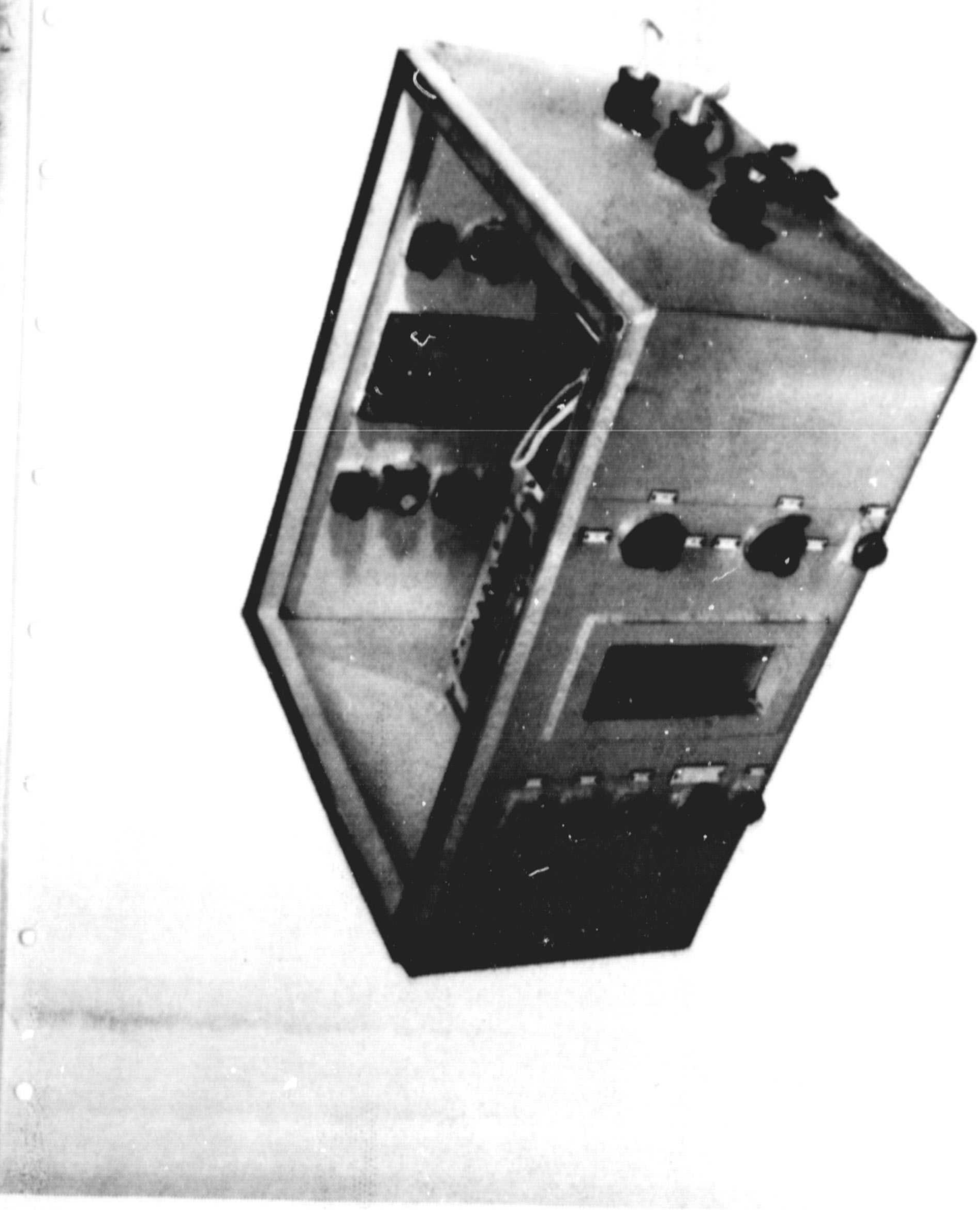


Figure 1.1.1 Explosion Proof Enclosure

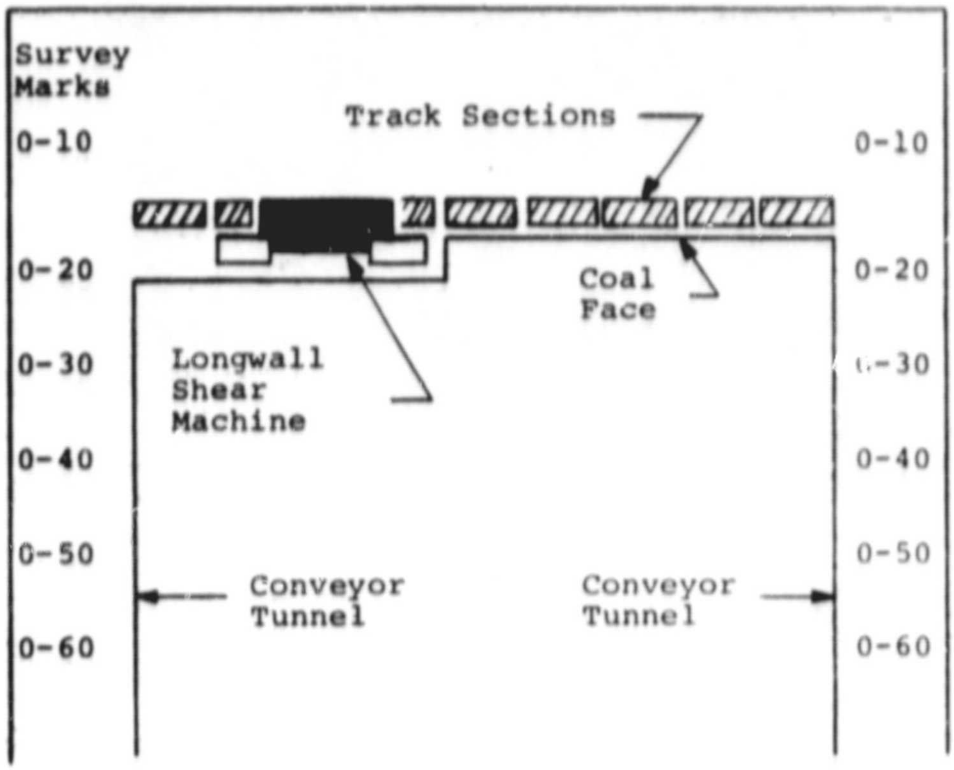


Figure 1.2 Longwall Mine - Top View

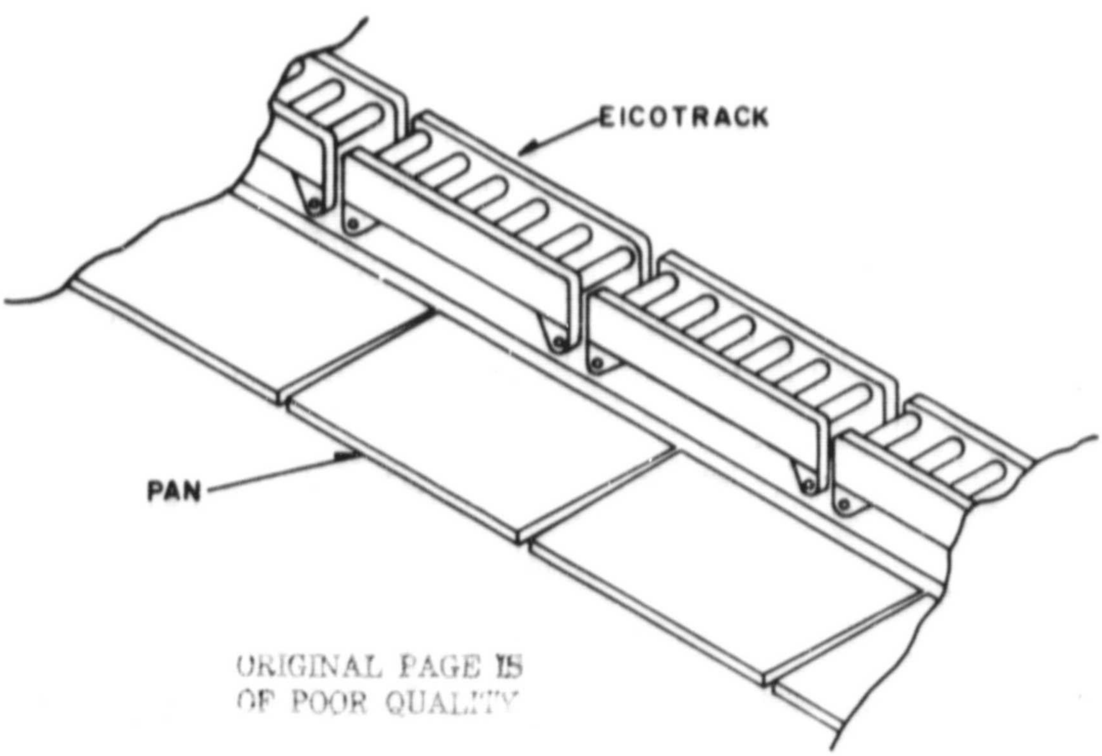


Figure 1.2.1 Track Sections

of the unit can total several hundred feet. The actual measurement surface is the "Eicotrack" rack face. It is located on the "gob" side of the conveyor which is the side away from the coal face. See Figure 1.2.1.

2.0 SYSTEM CONFIGURATION

2.1 General

A block diagram of the system is illustrated in Figure 2.1. It is divided into intrinsically safe and unsafe equipment. The safe equipment consists of two high accuracy resolvers, one incremental optical encoder and initiation switches. These pieces are housed in the transducer assemblies. The remaining equipment is housed within an explosion-proof enclosure.

2.2 Angle Measurement

The angle transducers are mounted in a mechanical support structure called the "angle cart". This unit contains the mechanics that allow the transducers to accurately produce repeatable angular positions. The angle transducers are intrinsically safe brushless resolvers and are directly connected to the angle measuring shoes. Figure 2.2 provides an exposed view of the "angle cart".

The angle between rack sections is measured differentially by the two resolvers. They are attached to a common reference

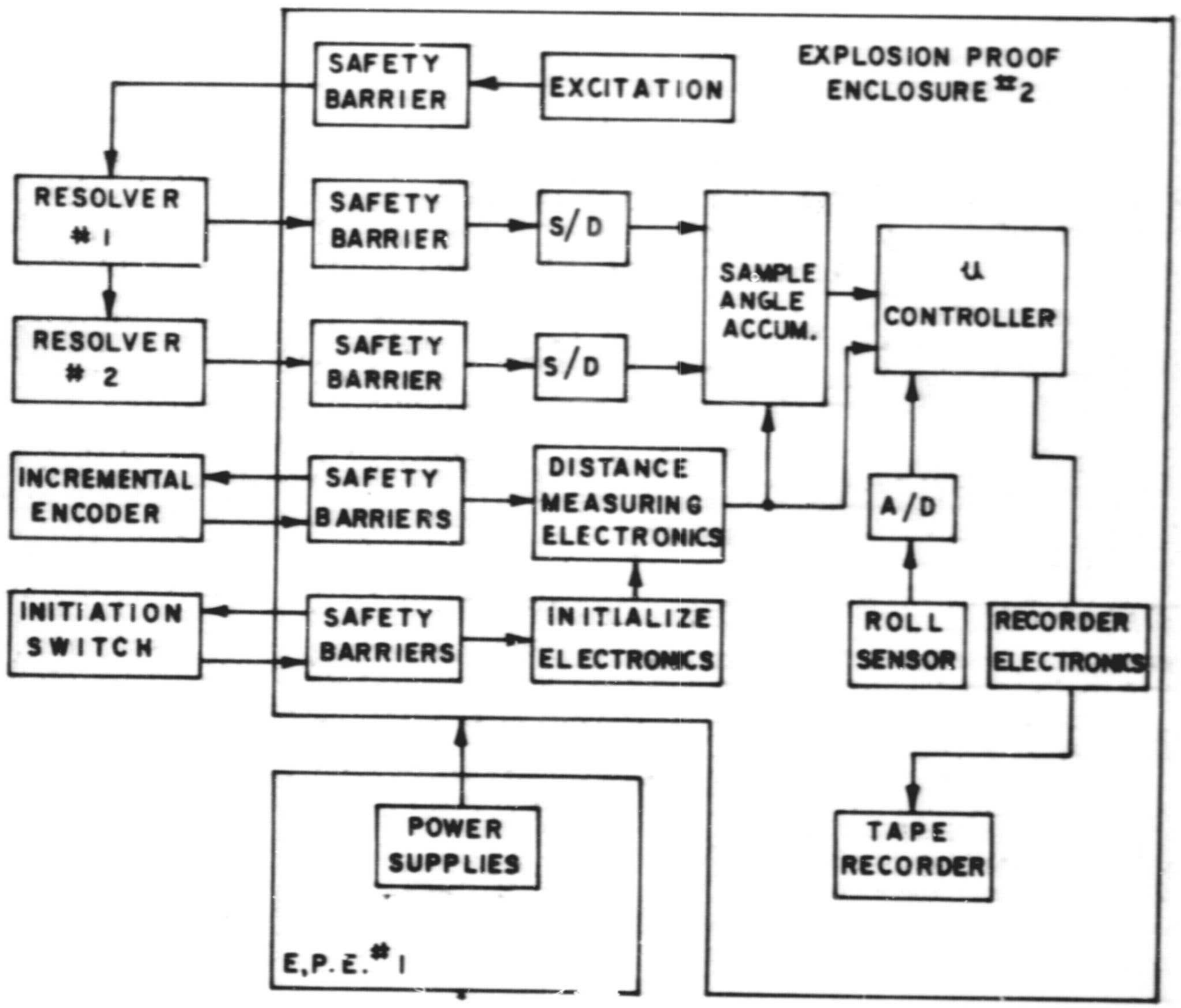
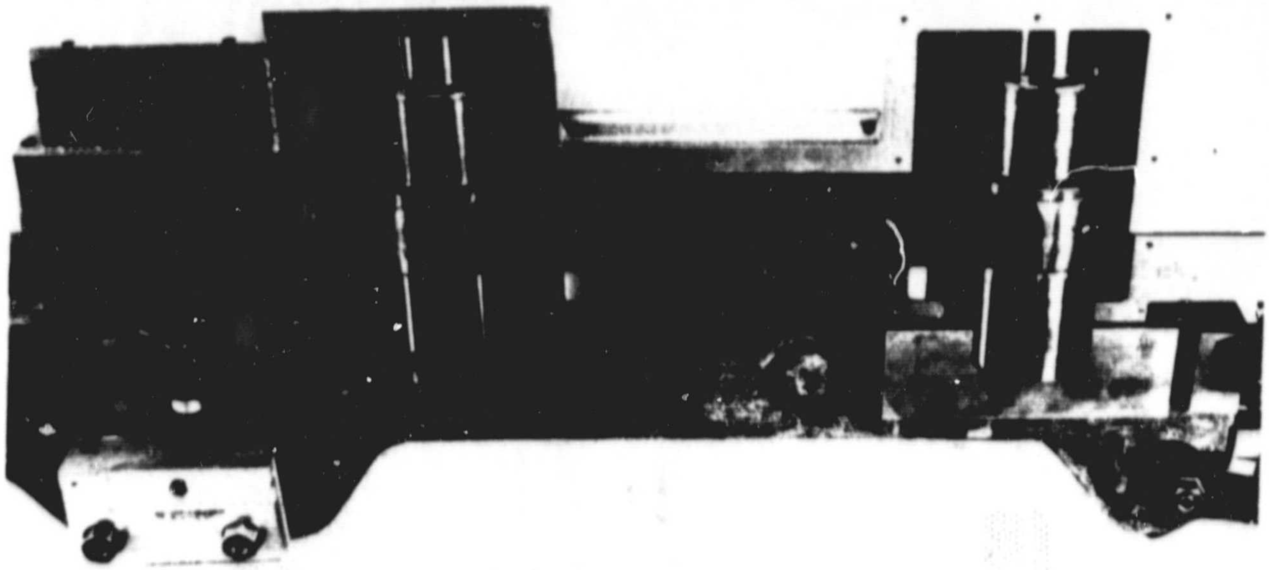
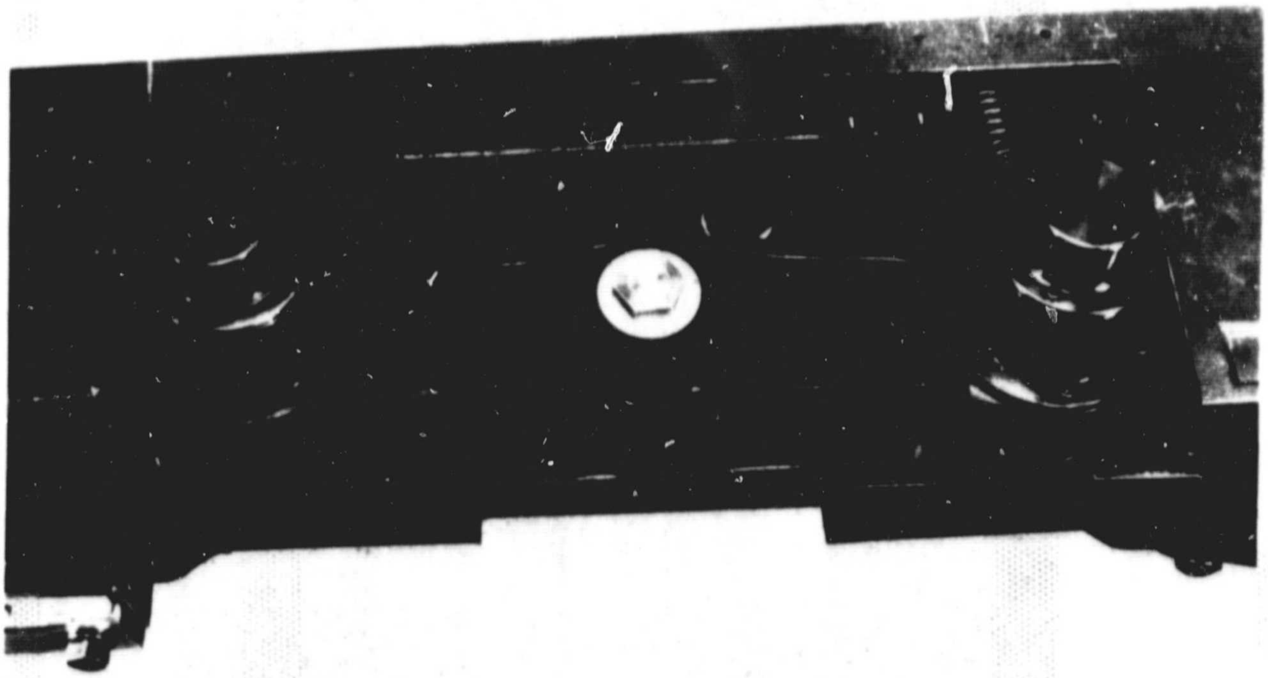


Figure 2.1 Measuring System Block Diagram



(a) Side View - Partially Uncovered



(b) Top View - Completely Uncovered

Figure 2.2 Angle Transducer Assembly

surface and are displaced from one another by a distance that is adequate to bridge two rack sections simultaneously. This relationship is illustrated in Figure 2.2.1.

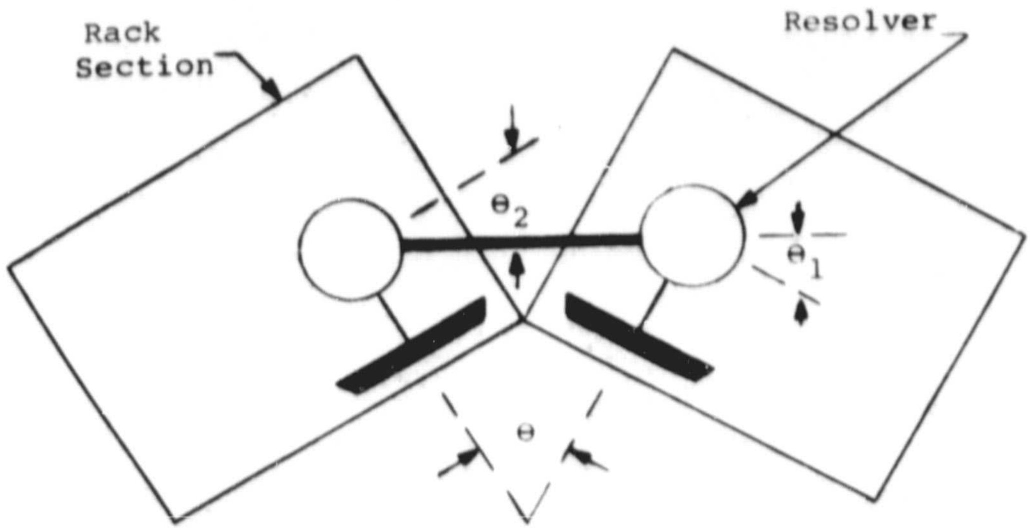


Figure 2.2.1 Track Angle Measurement Technique

The measuring device is designed so that θ, the angle between sections, is the sum of θ_1 and θ_2. This design permits the measured angle value to remain independent of the angle measuring device location.

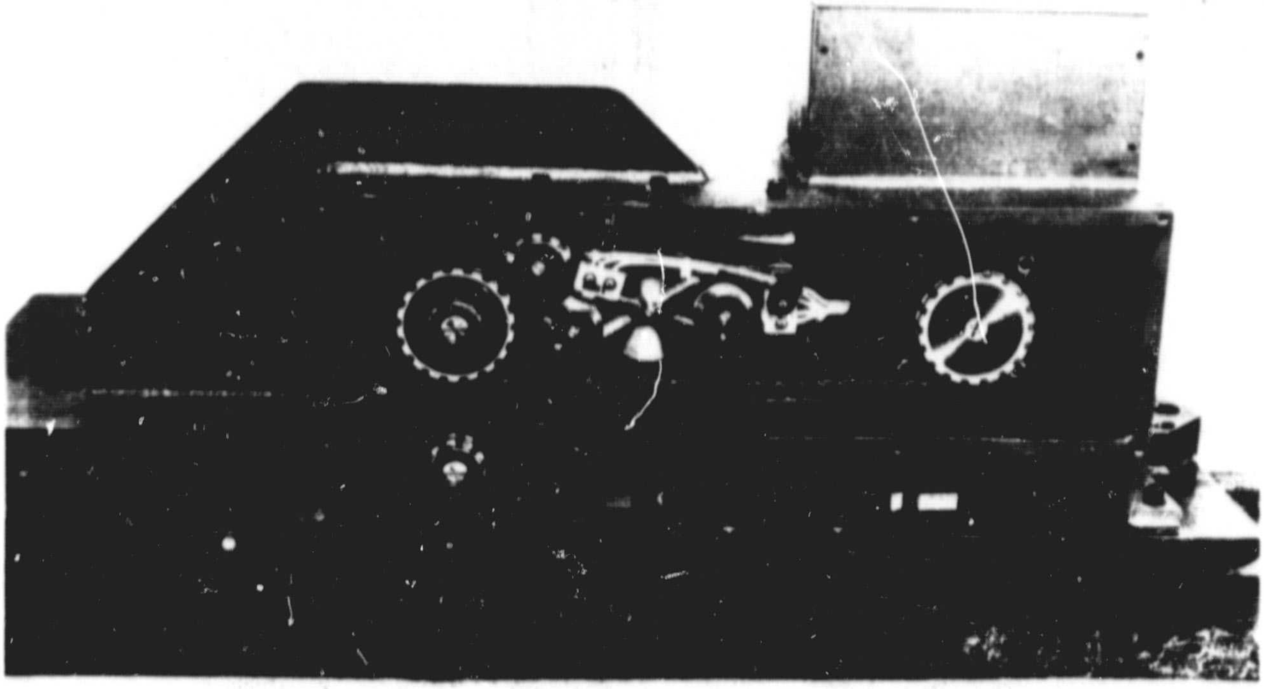
Wear and vibration may generate errors if the initial relationship between the two resolvers is disturbed. Vibration may produce a change which is static in nature and results in an error which occurs at a discrete interval of time. When it does occur, all angle measurements thereafter deviate from the correct value by the same fixed amount. Wear produces a change in the measurements which is gradual but cummulative. These two problems are handled by making a calibration measurement, across a surface that does not change, just prior to measuring a rack angle. This measurement is known as the "bias angle measurement" and is made along each rack section. Uncertainties that arise because of surface irregularities are handled by making a large number of bias measurements over a short distance of track rather than one measurement at a fixed point. In the computation phase, the bias readings are averaged and the result is subtracted from each angle reading to prevent any accumulative error.

2.3 Distance Measurement

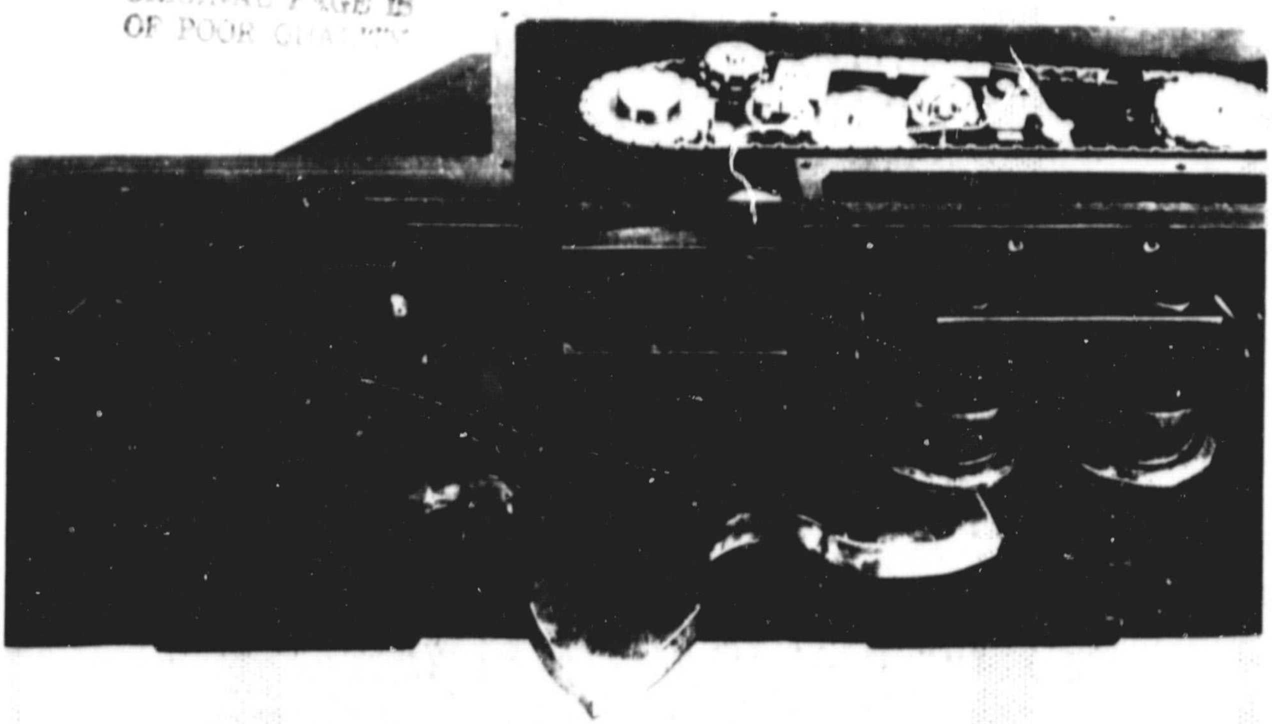
Distance measurement is comprised of defining 1) where the coal shearer is relative to a known starting location and 2) when bias and angle measurements should be made. The equipment needed to perform these functions consists of electronic detection and measurement circuitry operating in conjunction with an encoding transducer which is interconnected through gearing to the shearing machine drive rack.

Distance is measured by counting pulses that originate from a 500 line optical incremental encoder. The encoder is located inside a support structure called the distance cart which is mounted overtop the shearer drive rack. A five point starwheel-type gear contained in the distance cart makes contact with the rack and rotates one fifth revolution for each 126 millimeters of forward movement. The starwheel rotation is transferred to the encoder through a system of gears that permit distance measurement to be resolved to every 1260 micrometers of travel. Figure 2.3 illustrates the side and bottom views of the distance transducer assembly.

Before distance can be measured, a known reference point must exist. This point, known as the zero point, is defined by the simultaneous activation of limit switches and the encoder zero pulse. To guarantee that only one reference point occurs along a coal face the limit switches are geared to the starwheel so that one actuates for every 7.14285 rotations and the other for every 357.14285 rotations. This coarse/fine configuration guarantees that only one reference indication occurs for each 885 feet of shearer motion. Once the zero reference point has been defined, absolute distance is measured by an electronic up/down counter which accumulates pulses from the encoder as the shearer moves along the face.



(a) Side View Showing Limit Switches and Gearing



(b) Bottom View Showing Starwheel

Figure 2.3 Distance Transducer Assembly

The position at which bias and angle measurements are made is determined by a second set of electronics that operates in conjunction with the encoder and the absolute up/down counter. The initial point at which the first bias measurement occurs is defined by a comparator network output that goes high when the absolute position value is greater than a thumbwheel set value. As the comparator goes high, another counter network turns on and it is this counter's output which is compared with a distance measurement value, an angle start value and a distance measurement recycle value. The operation of the second counter is cyclic since the shearer rack geometry is cyclic.

2.4 Roll Measurement

The roll transducer is mounted within the explosion-proof enclosure. The transducer produces an electrical signal proportional to angular displacement relative to a vertical reference. The unit consists of a pendulum submerged in damping fuild. The pendulum's position is sensed to provide an analog voltage output of one half volts per degree of inclination. The analog output signal is converted into digital format by a twelve bit D/A (digital-to-analog) converter. The most significant bit weight is 16 degrees and the least significant bit weight is .0078 degrees.

3.0 SYSTEM OPERATION

3.1 General

The measurement system is designed to operate as the longwall shearer makes its numerous passes across the coal face. During this time the hardware electronics is sending transducer readings to the system microprocessor (Rockwell ATM 6500) for temporary storage. Each time the machine moves to the end of the conveyor track the accumulated data is stored on a cassette tape. After this transfer the same temporary data is reduced mathematically by computer firmware (non-volatile software) into a useable format for display.

During the calculation process the present data is reduced with data acquired when the shearer was first installed in the mine. Normally before the first run the conveyor track is aligned as straight as possible. Then manual measurements are made of the conveyor track to obtain the exact alignment. The resulting "manual data" is put on a non-volatile integrated circuit memory chip which is added to the system to become a part of the firmware. Afterwards the shearer traverses the total coal face to obtain angle information defined as the "initial run data". Subsequently this and the "manual data" becomes the basis of all the system calculations.

The system firmware performs the following:

1) Initialize system
2) Input and store transducer data
3) Transfer accumulated data to tape
4) Reduce accumulated data to a display format
5) Display contour of longwall conveyor
6) Provide information on multiple shearer parameters.

A general system flow chart is given as Figure 3.1.

3.2 Power-up and Initialization

At power-up the computer zeroes all of the data and control registers. It then configures each input/output port for correct data flow. The computer also sets numerous control registers and retrieves the conveyor "manual" data from non-volatile memory for later use. After set-up the computer begins to convey visual information to the machine operator. The first display reads "At headgate position push start button". When the operator acts upon this instructive command the distance and angle counters are zeroed and the acquisition of data is enabled.

The operator may alternately set the system for an "initial run" by setting the initial run (key-type) switch to "On" before power-up. When this is done the initialization routine is the same except that the cassette tape is also positioned to its starting point.

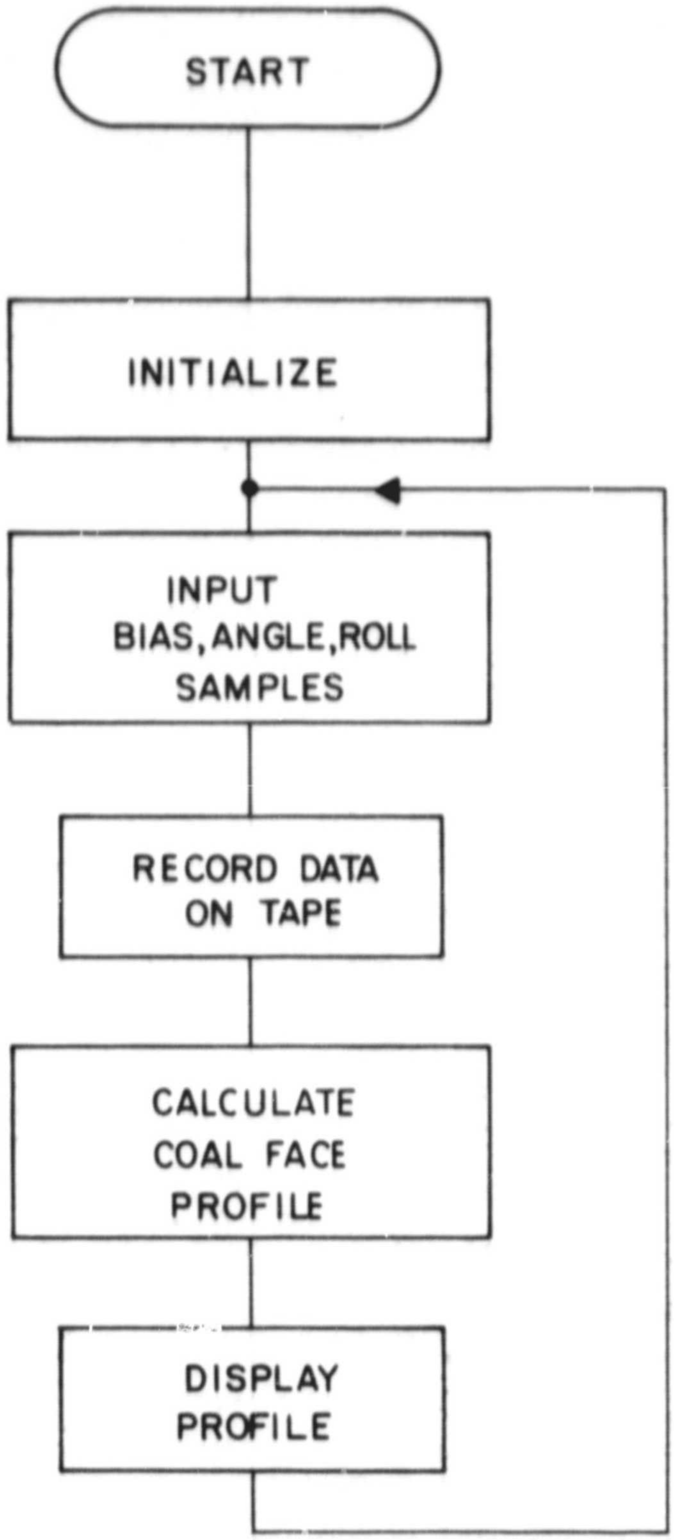


Figure 3.1 System Flow Chart

3.3 Data Acquisition

On the physical system the resolvers are displaced by fifteen inches from one another. Each bias measurement is then made across a five inch span of the thirty inch trackage starting at a point that is three and one half inches from the end for the trailing transducer and eighteen and one half inches for the leading transducer. Measurement is complete when the trailing transducer has moved eight and one half inches from the track section end - and the leading transducer has moved to the twenty-three and one half inch location. An angle measurement is initiated when the trailing transducer has moved to the eighteen and one half inch location and the leading transducer has moved onto the next track section three and one half inches. It is completed when the trailing transducer has moved an additional five inches to twenty-three and one half inches and the leading transducer to the eight and one half inch point. Figure 3.3 illustrates the technique used to make these measurements.

The procedure that is used to obtain transducer readings is primarily determined by the distance and angle electronics. In this electrical section the following are switch selectable:

1) Number of angular samples
2) Start location of first bias measurement
3) Start location of angle measurements
4) Measurement cycle repetition distance

See Figure 3.3.1.

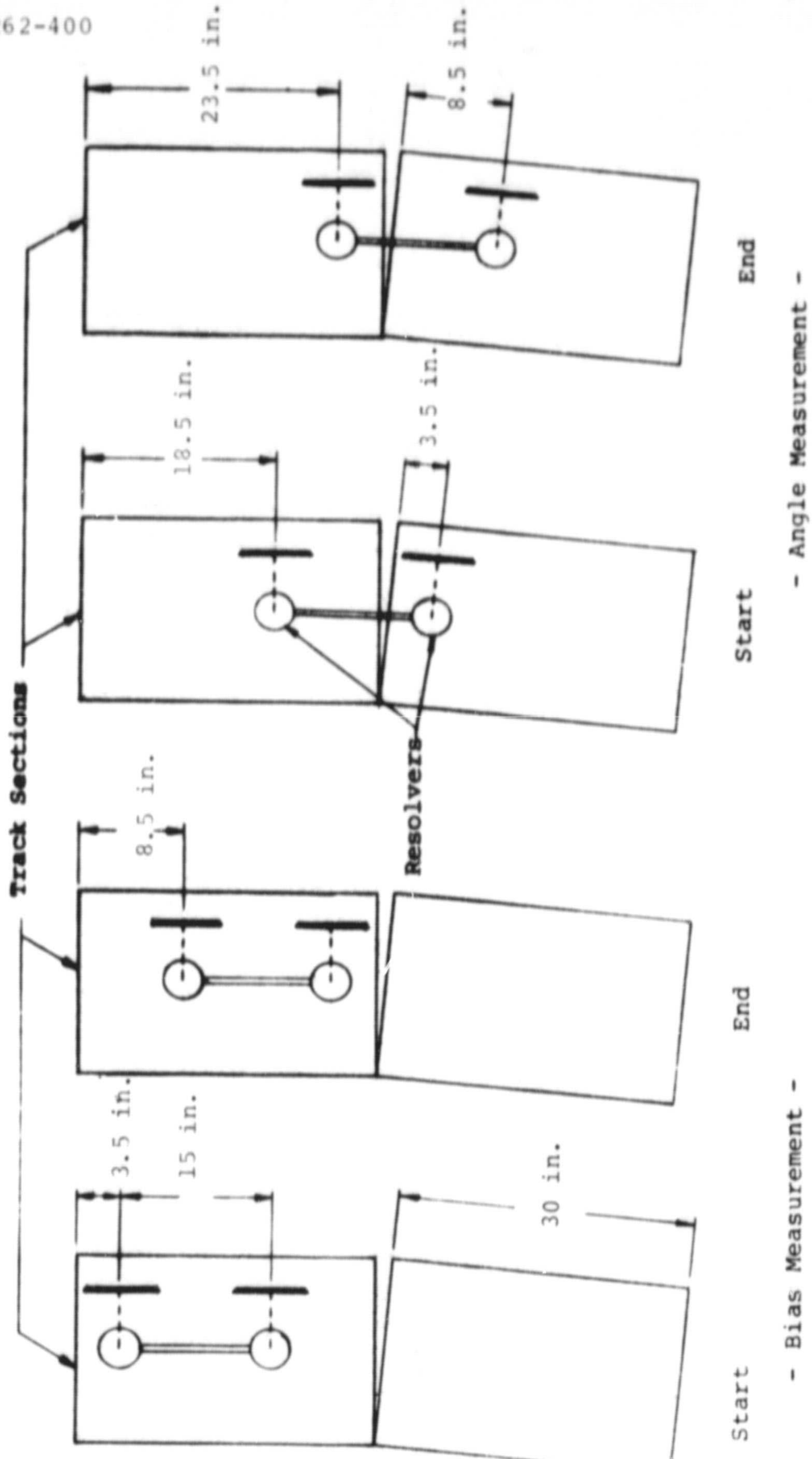


Figure 3.3 Bias and Track Angle Measurement Technique

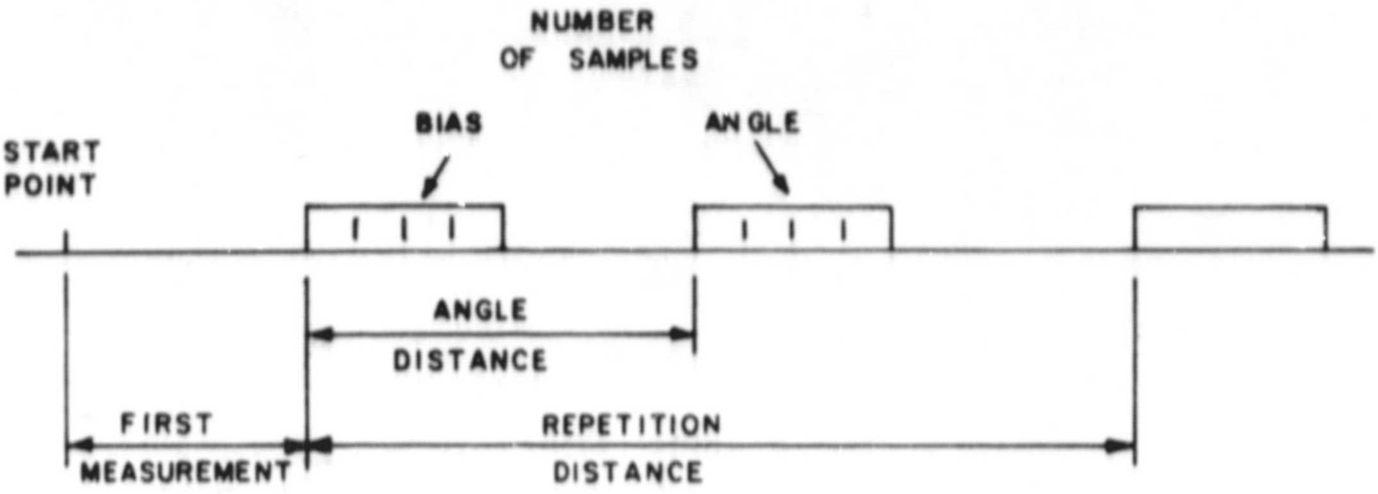


Figure 3.3.1 Data Acquisition Sequence

The electrical hardware performs two major functions. First it keeps track of the linear distance traveled by the mining machine from the start point. The second use is to obtain and add up the 100 measurement samples for each angle. After addition the data is temporarily held and the computer is sent an interrupt signal.

Upon receipt of this signal the data is immediately transferred to temporary memory for future application. The computer then takes a roll measurement, saves the result and then waits for the next interrupt.

3.4 Data Storage

After making readings on the last conveyor track section the system transfers to tape the accumulated bias, angle, and roll measurements. This data is stored on a digital cassette recorder in serial phase encoded (Bi-phase-level) format. The recorder used to store this information is a Raymond Model 6406 "Raycorder". This serves as the long term, non-volatile, bulk data storage device for this project.

The information is retained on tape in the form of files. Each file corresponds to one "run" of the mining machine; where a "run" is the action of cutting the complete coal face once. A 300 foot cassette tape was selected and can retain the runs generated over a one month period.

If the mining machine has just completed an initial run, then this original data is also transferred to non-volatile memory for future use.

3.5 Data Reduction

The data reduction consumes the most time of the many system activities. As an example; a mining machine with a conveyor track 600 feet long would require approximately two minutes of system data processing. The numerous calculations that are made by the system computer are outlined as a general flow chart in Figure 3.5.

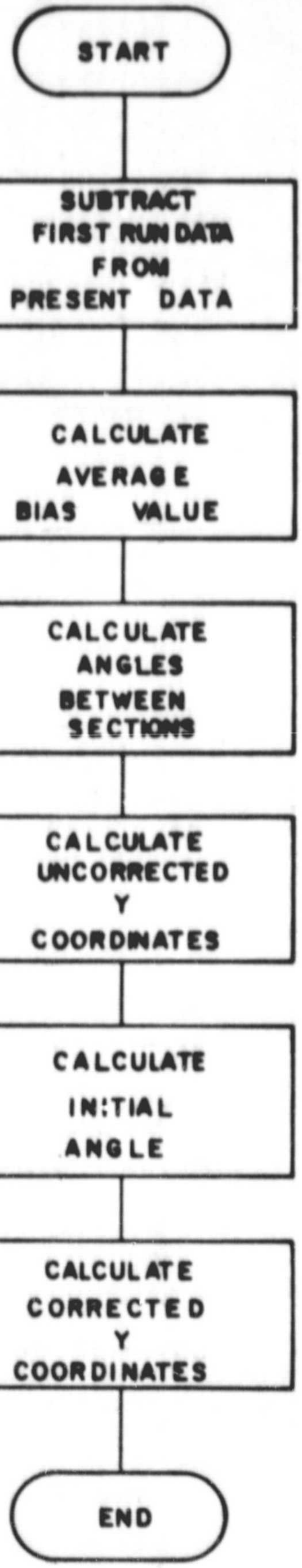


Figure 3.5 Data Reduction Flow Chart

After the data has been converted to decimal values the program calculates the bias deviation for each reading. This is done by comparing the present bias measurements with the "initial run" bias measurements. Under normal conditions this difference would be very close to zero. As soon as this calculation has been performed for each value then the average bias deviation is determined. The combined computation performed is:

$$B = \left(\sum_{i=1}^{N} (B_i - B_i \text{ init}) \right) / N$$

where:

B - average bias deviation
B_i - present bias value
B_i init - initial run bias value
N - number of angles measured

The next program segment calculates the angle change across each rack section by summing the manually measured data to the difference value; the difference value being the deviation of the present angle measurement from the initial run angle value. The computation performed is:

$$A(i) = (A_i \text{ pres} - A_i \text{ init}) + A_i \text{ man}$$

where:

A (i)	-	corrected angle value
A_i pres	-	present angle value
A_i init	-	initial run angle value
A_i man	-	manually measured angle value

With the results from the two preceeding equations we can calculate the actual angles between each rack section. The equation used is:

$$A(I) = \sum_{j=1}^{I} \sum_{i=1}^{j} (A(i) - B)$$

where:

A (I)	-	referenced angle value
A (i)	-	corrected angle value
B	-	average bias value

Figure 3.5.1 clearly shows the summation process.

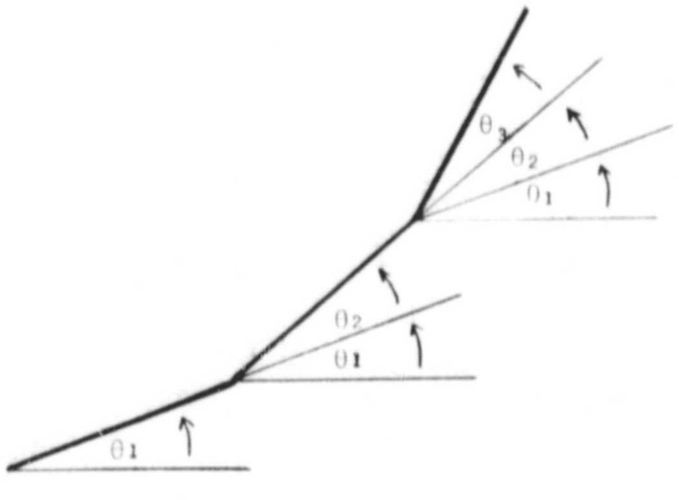


Figure 3.5.1 Graphic Summation of Measurement Angles

At this time the firmware determines the 'Y' coordinate for each rack section referred to the position of the first rack. The general equation is:

$$Y(j) = Y(j-1) + L * SIN\left(\sum_{I=1}^{j} A(I)\right)$$

where:

- Y (j) - uncorrected Y value
- L - length of rack section
- A (I) - referenced angle value

Figure 3.5.2 presents a simple example. Y (Ø) equals zero. Y (1) equals zero because the initial angle is assumed zero.

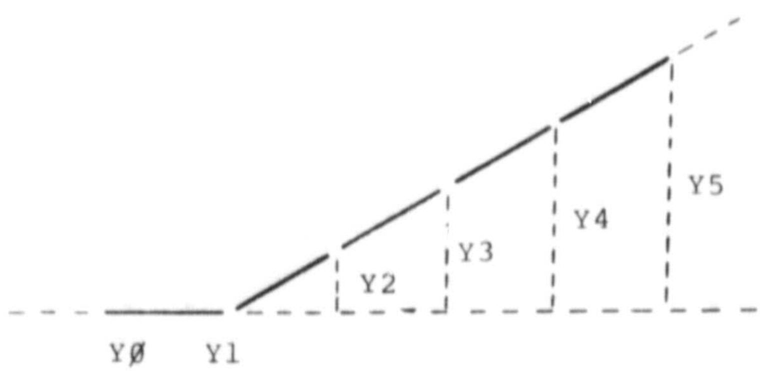


Figure 3.5.2 Uncorrected Y Coordinate Graph

Once the last uncorrected Y coordinate value is determined the conveyor track can be referenced to the position of the zero rack section. This is done by calculating the initial angle. In the following explanation Figure 3.5.3 may be used as an aid.

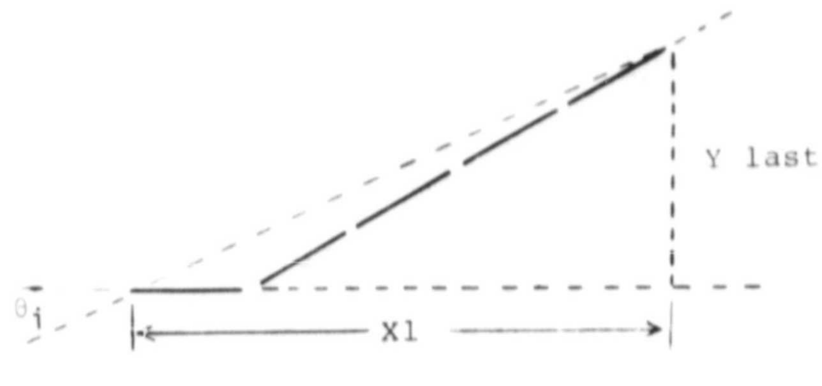


Figure 3.5.3 Graphic Initial Angle Determination

The initial angle would be derived most accurately by using:

$$\theta_i \text{ (actual)} = \arctan (Y \text{ last}/X1)$$

where:

θ_i (actual) - accurate initial angle value

Y last - last uncorrected Y coordinate value

X1 - accurate X distance

A very close approximation to this would be:

$$\theta_i \text{ (practical)} = \arcsin \text{ (Y last/(L*N))}$$

where:

θ_i (practical)	-	practical initial angle value
Y last	-	last uncorrected coordinate value
L	-	length of rack section
N	-	number of rack sections

The arcsine equation was used because it does not require the system to calculate all of the X coordinates and thus saves on computer processing time. Also the angles derived from each equation were very similar. The equation used by the computer is:

$$R = \arcsin \left(\frac{Y(\emptyset) - Y(N)}{L * N}\right)$$

where:

R	-	initial value
Y (Ø)	-	Y value for rack zero
Y (N)	-	Y value for last rack
L	-	length of rack section
N	-	number of rack sections

The last major calculation that the microprocessor performs is to determine the Y coordinates for each rack section relative to the initial rack section.

The equation is:

$$Y(J) = Y(J-1) + L * SIN \left(\sum_{I=1}^{J} (A(I)+R) \right)$$

where:

Y(J) - corrected Y value

L - length of track section

A(I) - referenced angle value

R - initial angle

The resulting coordinate relationship is represented by

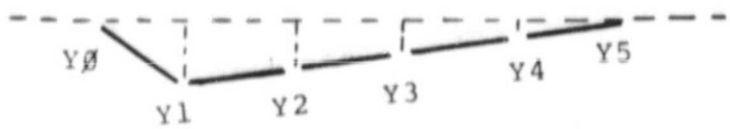


Figure 3.5.4 Corrected Y Coordinate Graph

3.6 System Display

The coal face measurement system provides the longwall machine operator with two useful display routines. The first displays the general curvature of the conveyor track while the second pinpoints particular physical parameters of the mining machine.

As the system operates, the curvature of the conveyor track is automatically displayed each time the mining machine cuts the complete coal face. This curvature is specified as the physical displacement of the track from an imaginary center-line stretching from one end of the conveyor track to the other. The length of the track is divided into 15 equal segments which allows for a track contour displacement to be presented for 15 locations. The displacement is in feet and the direction of the curve is either toward the coal face (FACE) or away from the coal face (GOB). The operator sees an alternating display showing the displacement to either side of the center line. The display alternates every four seconds. See Figure 3.6.

Ø	1	1	2	2	3	2	1	Ø	-	-	-	-	-	Ø	FACE
Ø	-	-	-	-	-	-	-	Ø	1	2	3	2	1	Ø	GOB

Figure 3.6 Conveyor Displacement Display

During a mining operation the operator can also obtain the following system parameters:

1) Previous displacement of any rack
2) Present roll of the mining machine
3) Roll of any rack previously traversed
4) Actual distance of the mining machine from the starting point

These functions may be called by first momentarily pushing the "function" pushbutton on the XP enclosure. See Figure 3.6.1. This permits the computer to enter the display routine and present instructions to the operator. By following the displayed directions, the operator is lead step by step through the display format. Figure 3.6.2 through 3.6.4 describes the computer activites, the information displayed and the actions required of the machine operator.

The system also signals the shearer operator for any of the following possible conditions:

1) Power failure
2) Resolver angular error
3) Cassette loading error
4) Cassette tape error
5) Cassette tape full

A resolver error is flagged when the computer senses that an angle reading is ten degrees or more. If this occurs the system displays question marks to insure that the operator does not act on erroneous data. A cassette loading error is signalled when a cassette has not been inserted and/or the recorder door has not been closed. A cassette tape error occurs whenever the recorder determines that there was an error in the transfer of data to the tape.

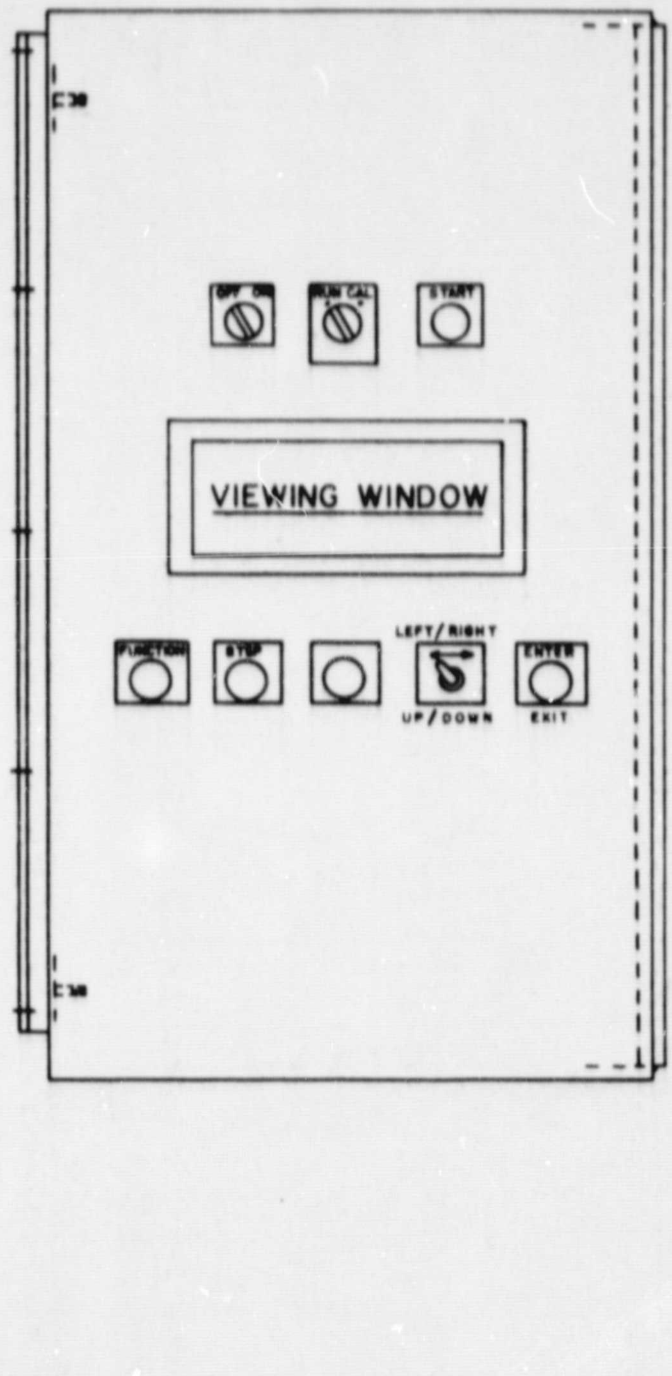


Figure 3.6.1 XP Enclosure

I. ENTRY TO ROUTINE

ACTION	RESULT
Momentarily Push Function Button	Displays "DISPLAY ROUTINE", Delays 2 sec. Displays "PUSH FUNCTION BUTTON", Delays 2 sec. Displays "& THEN ENTER BUTTON", Delays 2 sec. Loops and displays again.

II. POSSIBLE FUNCTIONS

Push and Hold FUNCTION Button	Displays "DISPLACEMENT OF RACK", Delays 2 sec. Displays "PRESENT ROLL", Delays 2 sec. Displays "ROLL OF RACK SECTION", Delays 2 sec. Displays "PRESENT DISTANCE", Delays 2 sec. Displays "EXIT FROM ROUTINE", Delays 2 sec. Loops and displays again.

III. SELECT FUNCTION

A. Release FUNCTION Button when desired function is displayed.	Displays "DISPLACEMENT OF RACK" or Displays "PRESENT ROLL" or Displays "ROLL OF RACK SECTION" or Displays "PRESENT DISTANCE" or Displays "EXIT FROM ROUTINE".
B. Push and Hold ENTER Button	Displays answer or requests information.
C. Release ENTER Button	

Figure 3.6.2 Displayed Parameter Entry Sequence and Result (Part I)

IV. FUNCTION 1 (DISPLACEMENT OF RACK) OR FUNCTION 3 (ROLL OF RACK SECTION)

ACTION	RESULT
(To be performed after selection) A. Momentarily Push STEP Button	Displays "PUSH STEP FOR RACK #" Displays "USE LEFT/RIGHT SW.", Delays 4 sec. Displays "TO SELECT DIGIT", Delays 4 sec. Displays "PUSH ENTER WHEN OK", Delays 4 sec. Turns on Light Emitting Diode over 100's digit.
B. Push and Hold STEP Button	Displays numbers which increment every second.
C. Release STEP Button	No longer increments numbers.
D. Push and Hold LEFT/RIGHT switch to either side.	Turns on Light Emitting Diode over successive digits.
E. Release LEFT/RIGHT Switch	Light Emitting Diode designates digit which may be incremented.
F. Push and Hold ENTER Button	If the entry is > number of rack sections then Displays "ENTRY TOO LARGE" and returns to beginning of function, otherwise it Displays "USE UP/DOWN SWITCH", Delays 4 sec.
G. Release ENTER Button	Displays "TO CHANGE RACK NUMBER", Delays 4 sec. Displays "PUSH EXIT TO EXIT", Delays 4 sec. Displays "& FUNCTION TO RETURN", Delays 4 sec. Displays answer.
H. Push and Hold UP/DOWN switch to either side	Increments or decrements rack section number.
I. Release UP/DOWN switch	Displays answer for appropriate rack section.
J. Push and Hold FUNCTION Button	Returns to allow new function selection.
K. Release FUNCTION Button	Selects function.
L. Push EXIT Button	Exits from routine and displays previous general curvature.

Figure 3.6.3 Displayed Parameter Entry Sequence and Result (Part 2)

V. FUNCTION 2 (PRESENT ROLL) OR FUNCTION 4 (PRESENT DISTANCE)

ACTION	RESULT
(To be performed after selection)	Displays "PUSH EXIT TO EXIT", Delays 4 sec. Displays "& FUNCTION TO RETURN", Delays 4 sec. Displays answer.
A. Push and Hold FUNCTION Button	Returns to allow new function selection.
B. Release FUNCTION Button	Selects function.
C. Push EXIT Button	Exits from routine and displays previous general curvature.

VI. FUNCTION 5 (EXIT FROM ROUTINE)

(To be performed after selection)	
A. Push EXIT Button	Exits from routine and displays previous general curvature.

VII. RESULTS

RESULTS	DISPLAY
FUNCTION 1 (Previous Displacement of any rack)	RACK_XXX_F_X.XX_FEET RACK_XXX_G_X.XX_FEET
FUNCTION 2 (Present Roll)	ROLL_F_XX.X_DEGREES_ ROLL_G_XX.X_DEGREES_
FUNCTION 3 (Previous roll of any rack)	RACK_XXX_F_XX.X_DEG_ RACK_XXX_G_XX.X_DEG_
FUNCTION 4 (Present Distance)	LOCATION_XXX.X_FEET_

Figure 3.6.4 Displayed Parameter Entry Sequence and Result (Part 3)

3.7 Data Retrieval

Each time the coal face is completely cut the accumulated readings are saved on the cassette tape for future evaluation. Thus after approximately one month a new cassette must be installed in the system. For protection from any contaminants, the cassette recorder mechanics is contained within a separate sealed box. So instead of switching tapes, boxes will swapped. In order to retrieve the data from the tape an identical Raymond cassette recorder should be used. And with the appropriate Raymond formatter option, direct input to a computer is accomplished.

4.0 TEST RESULTS

4.1 General

The accuracy specification for the yaw measurement equipment required that the overall system be capable of calculating the true curve for a 600 foot coal face to within ± 12 inches of its true position.

System accuracy was confirmed through the performance of nine tests at the Bruceton facility. These tests included measurement across a straight track, a track bowed approximately two feet towards the coal face and a track bowed approximately one foot away from the coal face. All testing was performed utilizing fifteen Eickhoff pan sections to obtain a representative longwall configuration. Figure 4.1 illustrates the straight face track geometry at the Bruceton facility.

4.2 Test Procedure

The initial test to confirm system performance was undertaken on a straight track. The test results were positive. The straight track tests were rerun two additional times to confirm repeatability. Again the results were very good. For the series of measurements the maximum deviation between the manually measured values and the system measured values was .03 inches. See Figure 4.2.

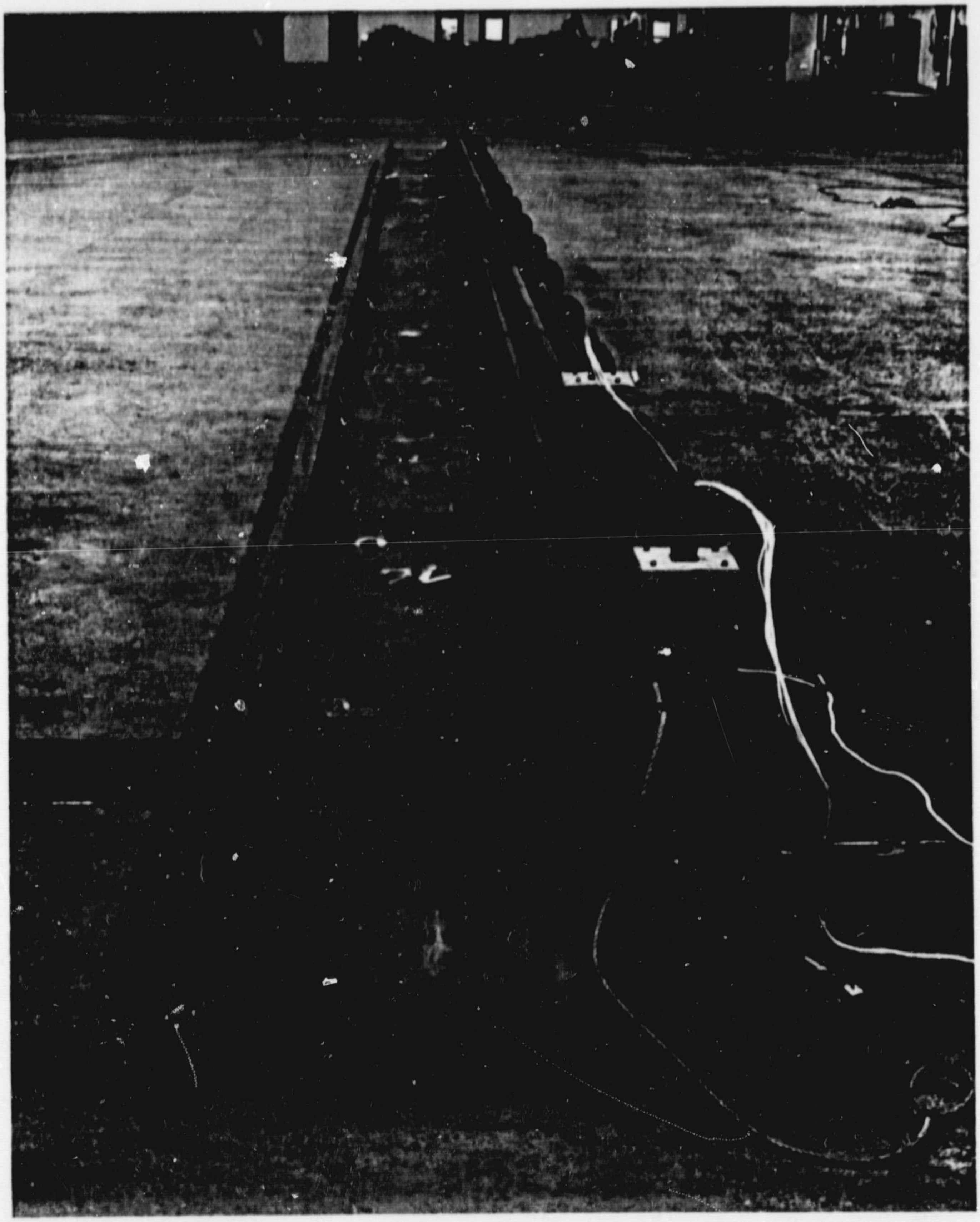


Figure 4.1 Test Facility

The track was next bowed approximately two feet towards the coal face. The bow was placed about midway between the end points in an attempt to generate large positive and negative angle readings. Three tests were made and compared with the actual track curvature. The actual and computed results were the same to within .29 inches. Figure 4.2.1 illustrates the track geometry and the computed results.

The third configuration was a bow approximately one foot away from the coal face. The results between computed and actual geometries were the same to within .11 inches. Figure 4.2.2 illustrates the track geometry and the computed results for this configuration.

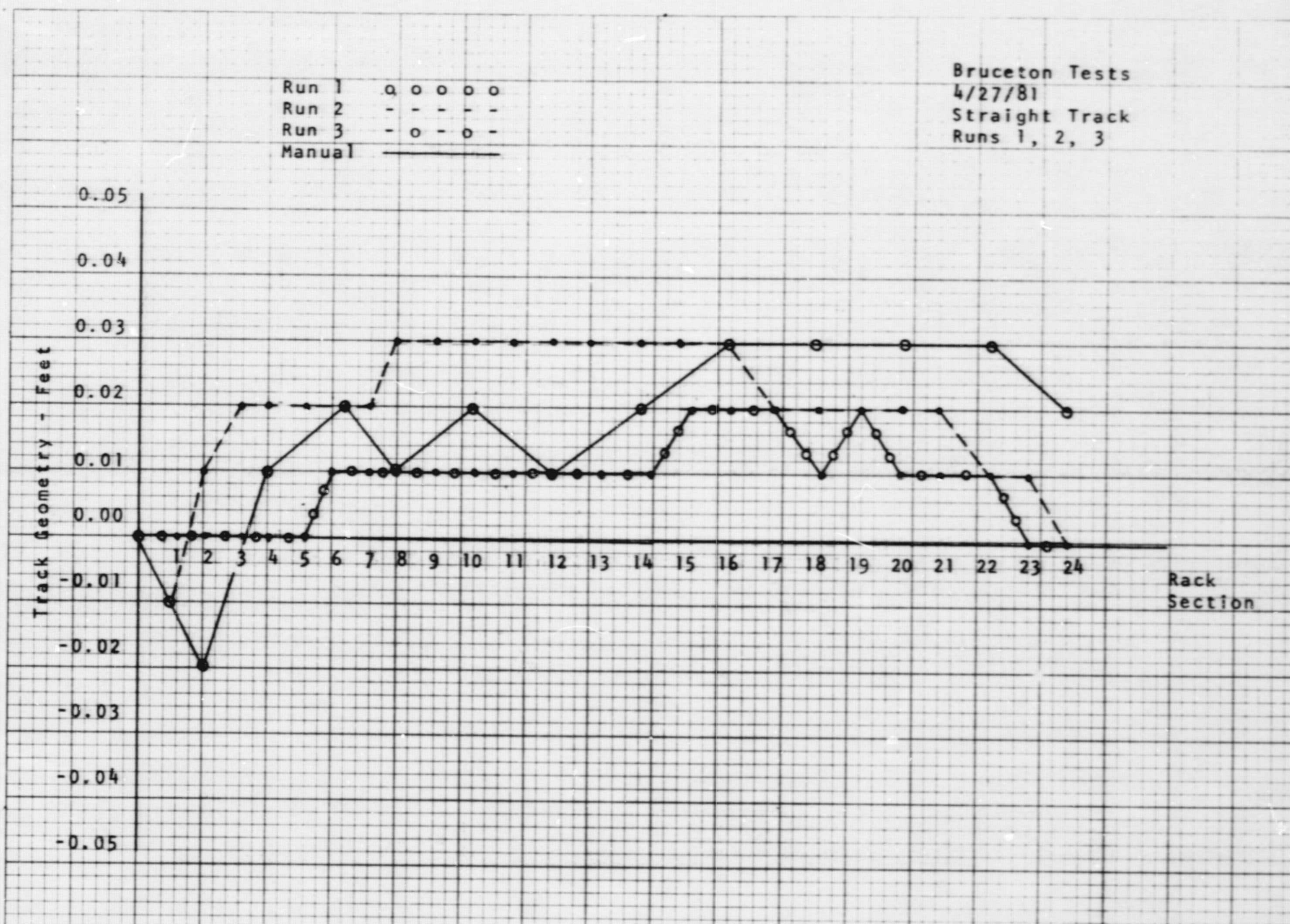


Figure 4.2 Straight Track Contour

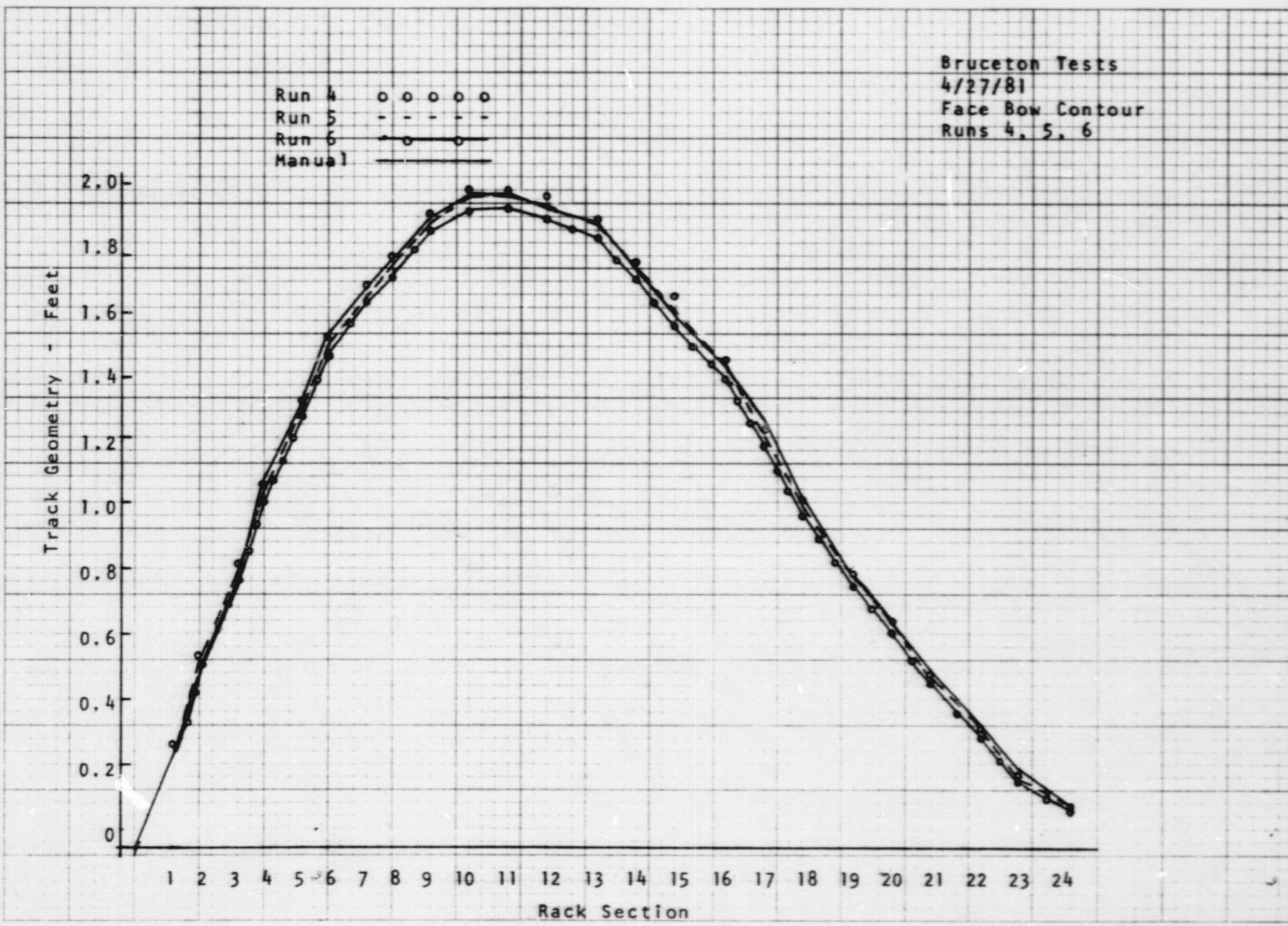


Figure 4.2.1 Face Bow Track Contour

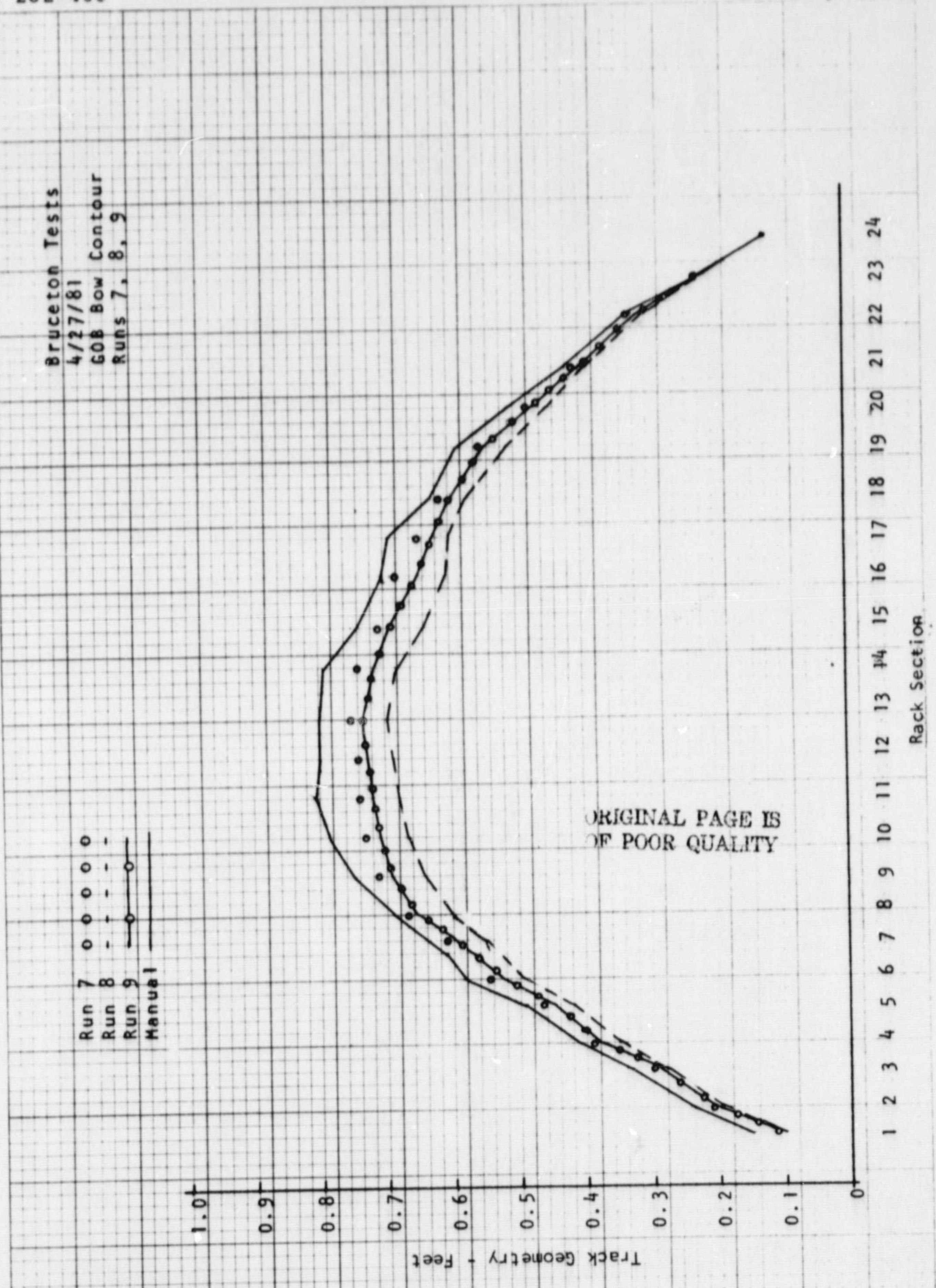


Figure 4.2.2 GOB Bow Track Contour

5.0 CONCLUSION

A microprocessor based contour measurement system adaptable to the Eickhoff shearer was developed for underground use.

The equipment was tested successfully at the Bruceton facility for a number of simulated coal face contours. Results indicate that the equipment can repeatedly measure shearer conveyor angles to within ± 0.05 degrees and contours to within 0.1 foot of the correct value.

Several measurement techniques were developed that bear promise for use with other mine related equipment. The resultant equipment includes intrinsically safe angle and distance measuring transducers.

The angle measurement system required that a non-contacting low energy transducer be acquired. A search resulted in the procurement of a low voltage 1 minute accurate size 11 resolver. Circuiting was developed for use with it and the simultaneous goal of accuracy and intrinsic safeness was achieved.

An incremental optical encoder was acquired, combined with energy limiting circuitry and limit switches to achieve an intrinsically safe absolute distance measuring system with a resolution of 0.05 inches.

The complete measurement system is presently under review by MSHA for compliance with intrinsic safety standards. When approval is received the equipment will be taken underground and tested for operability in a mine environment.

APPENDIX A

A.Ø COMPUTER OPERATION

A.1 General

The computer developed for the coal face measurement system consists of four integrated circuit boards. The AIM 6500 microcomputer makes-up the "heart" of the system. A memory board serves as the "brains" and two I/O boards allow the "heart and brains" to communicate externally. Figure A.1-1 presents the important features of each board.

The computer takes on the character of a measurement system with inclusion of operating programs. These programs are listed as Figure A.1-2 and are located on EPROM (Electrically Programable Read Only Memory) chips. They thus become a permanent part of the system.

In the following literature the programs are briefly discussed, shown as flowcharts, and completely listed. A system memory map is also provided.

1) AIM 6500 Microcomputer

4K	RAM
8K	BASIC
8K	MONITOR
20	Column LED Display

2) Memory Board

32K	Dynamic RAM
16K	PROM

EPROM Programmer

3) Input/Output Board

10	8 bit Ports per Board

Figure A.1-1 System Computer Hardware

PROGRAMS	LOCATION
INPUT/OUTPUT SETUP	4K EPROM
RUN NUMBER	
STARTING POINT	
CONTOUR DISPLAY	
ARC-TANGENT	
EPROM PROGRAMMER	
EPROM READER	
REGISTER SETUP	
BASIC	4K EPROM plus 2K EPROM
INPUT/STORE	2K EPROM
DATA TRANSFER	

Figure A.1-2 System Computer Programs

A.2 Program Execution Sequence

Immediately after turn-on all of the computer input/output ports are configured for system operation. After this the computer jumps to the BASIC routine which temporarily jumps to a normalization routine. When complete, the computer returns to the BASIC routine which signals the operator when ready.

While the coal is being cut the system is continually moving between the BASIC and Data Handling routines. The Data Handling routine inputs and stores the generated bias, angle, and roll data. At the end of each run, after all data has been input; the accumulated data is transferred to tape.

After replication the data is reduced algebraically to a useable form by the BASIC routine. The computer jumps to the Contour Display routine and presents the calculated results to the system operator.

Figures A.2-1 and A.2-2 provide a very general program execution sequence.

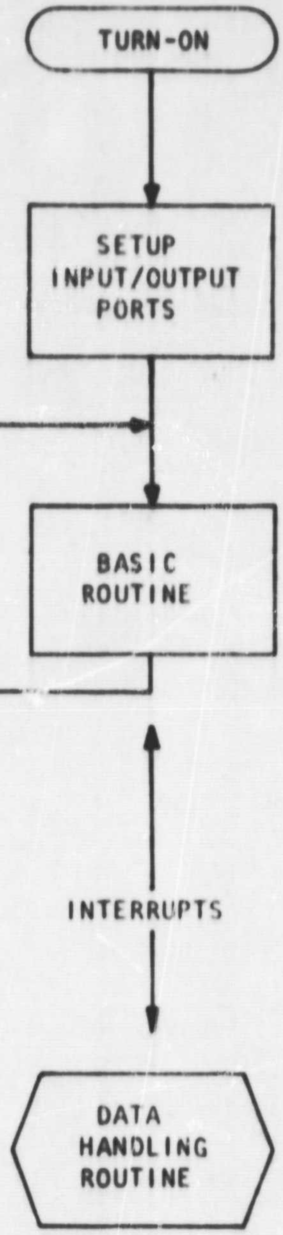


Figure A.2-1 General System Flowchart

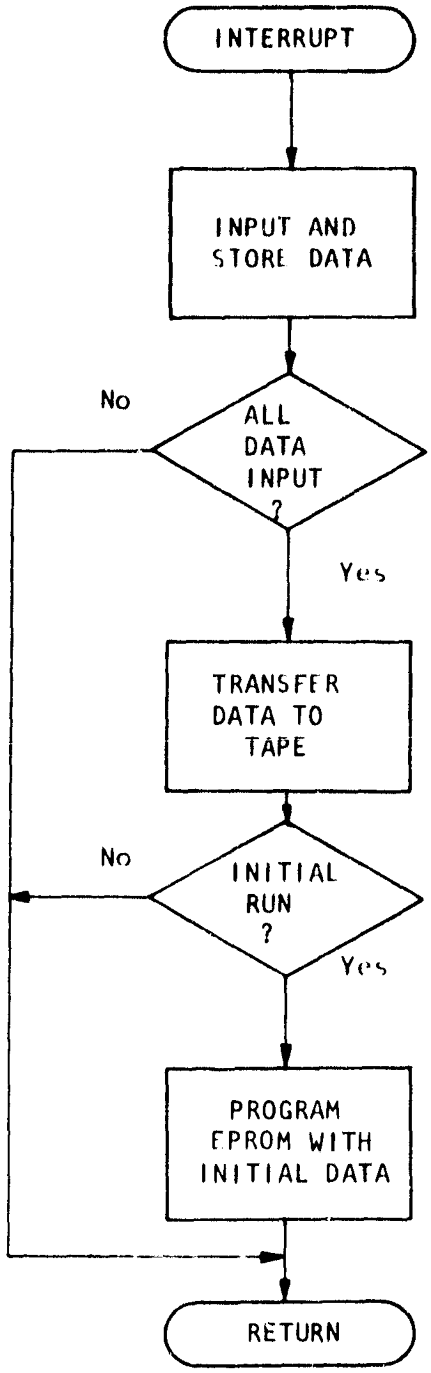


Figure A.2-2 Data Handling Flowchart

A.3 Input/Output Setup Program

The very first step that the computer makes is to send the tape recorder a stabilization command. This is done to insure that the tape does not move until required. After this all of the system input/output ports are configured for their particular application. The computer now executes the BASIC driver routine.

Figure A.3-1 shows the flowchart while Figure A.3-2 presents the actual program listing.

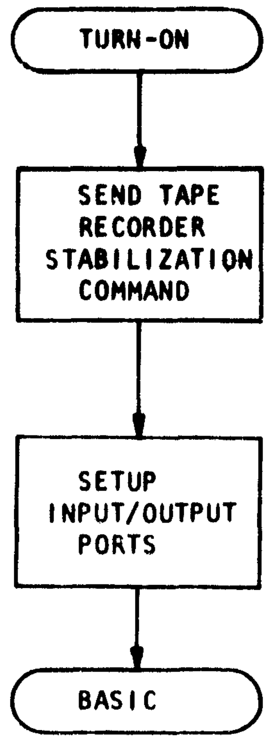


Figure A.3-1 Input/Output Setup Program Flowchart

```
*=$D000

----------------------
TAPE,PORTS,JMP

==D000
;DATA PORT REGISTER

==D000 PBC=$9502

;CONTROL REGISTERS

==D000 CRAC=$9501

==D000 CRBC=$9503

==D000 CRAD=$9601

==D000 CRBD=$9603

==D000 CRAL=$9D01

==D000 CRBL=$9D03

==D000 CRAM=$9A01

==D000 CRBM=$9A03

==D000 CRAS=$9B01

==D000 CRBS=$9B03

==D000 CRAT=$9F01

==D000 CRBT=$9F03

;DATA DIRECTION

==D000 DDAC=$9500

==D000 DDBC=$9502

==D000 DDAD=$9600

==D000 DDBD=$9602

==D000 DDAF=$9303

==D000 DDBF=$9302

==D000 DDAL=$9D00

==D000 DDBL=$9D02

==D000 DDAM=$9A00

==D000 DDBM=$9A02

==D000 DDAR=$9E03

==D000 DDBR=$9E02

==D000 DDAS=$9B00

==D000 DDBS=$9B02

==D000 DDAT=$9F00

==D000 DDBT=$9F02

;NORMALIZE TAPE UNIT

20EC   LDA #$00
DFA232 STA CRBC
20E1   LDA #$FF
DF20EC STA DDBC
DFA9   LDA #$04
008523 STA CRBC
850B   LDA #$57
==D011
850C20 STA PBC

;PORT SET, DIRECTION

A3E7   LDA #$00
B0FBAD STA CRAC
1EA4D0 STA CRAD
0DAD1C STA CRBD
A4853A STA CRAL
==D022
AD1DA4 STA CRBL
853B4C STA CRAM
31D0A5 STA CRBM
3BA63A STA CRAS
2042EA STA CRBS
20FADF STA CRAT
==D034
20A7E7 STA CRBT

;PORT SET FOR INPUT

B0FBAD STA DDAC
1EA4D0 STA DDAF
0DAD1C STA DDAM
A4853E STA DDBM
AD1DA4 STA DDAL
==D046
853F4C STA DDBL
52D0A5 STA DDAR
3FA63E STA DDBR
2042EA STA DDAT
A9E485 STA DDBT

;PORT SET FOR OUTPUT

3720   LDA #$FF
==D057
E9DF20 STA DDAD
48E8A2 STA DDBS

;PORT SET FOR OUT&IN

04BD   LDA #$0F
2EA495 STA DDBD

;PORT SET FOR DATA

A7CA   LDA #$04
10F820 STA CRAC
==D067
ECDFA2 STA CRAD
3F20E1 STA CRBD
DF2073 STA CRAL
E9C94E STA CRBL
D006A5 STA CRAM
370910 STA CRBM
==D079
853720 STA CRAS
ECDFA2 STA CRBS
4A20E1 STA CRAT
DF2071 STA CRBT

;JUMP TO BASIC

E8AD13 JMP $5000
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure A.3-2 Input/Output Setup Program Listing

A.4 BASIC Program

The main program utilized by the measurement system is the BASIC routine. All operations, if not performed as a part; originate from it. The program completes two major processes. One section calculates the general coal face contour from the accumulated data while another section displays more detail on the mining machine and its operation. The most recent addition to this program requests the operator to enter the mining machine's track end-point locations. This information references the measurement system results to a known coordinate geometry and thus closes the man-machine feedback loop.

Figure A.4-1 shows the flowchart while Figure A.4-2 through A.4-4 presents the actual program listing.

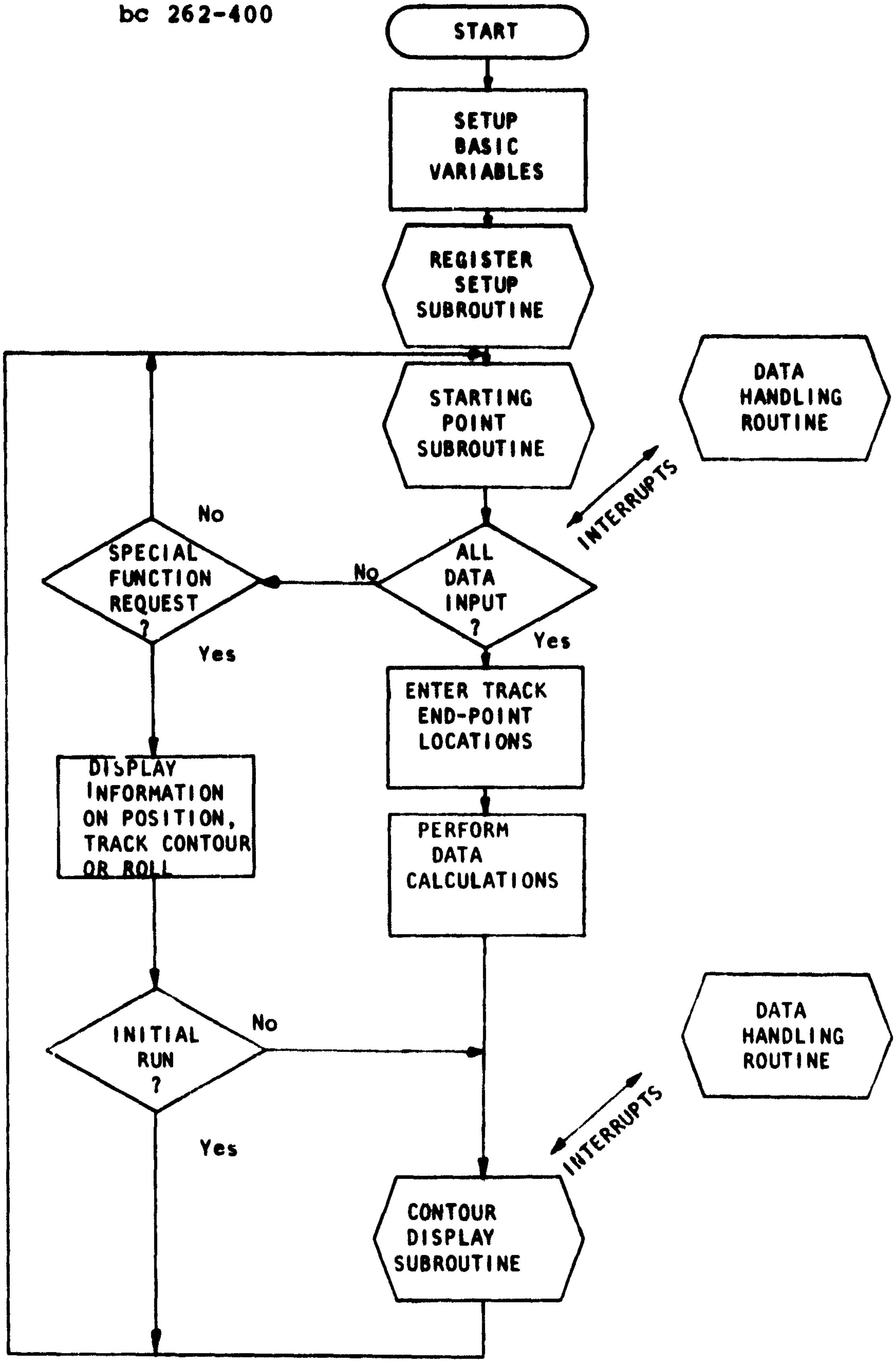


Figure A.4-1 BASIC Program Flowchart

```
LIST
 10 POKE4,80:POKE5,2
19:N=USR(0)
 20 M1=PEEK(17952):D
IMC(M1+1),D(M1+1),E(
M1+1),S(2)
 22 PRINT" ":PRINT"A
T HEADGATE POSITION"
:GOSUB1180
 25 PRINT" ":PRINT"P
USH START BUTTON":GO
SUB1180
 28 K3=.5:Z=0:IFPEEK
(37633)<>191THEN22
 30 POKE40460,236:PO
KE40460,204
 40 POKE40462,125
 50 POKE40461,130:PO
KE40462,130
 60 POKE40460,206:PO
KE40460,204
 70 K2=.0328:PRINT"
":IFPEEK(18433)<>0TH
EN80
 75 IFPEEK(18434)=0T
HENPRINT"INITIAL RUN
":GOTO90
 80 PRINT"PROCEED"
 90 POKE4,112:POKE5,
209:N=USR(0)
 100 IFPEEK(17677)=2
55THEN114
 110 IFPEEK(37633)=2
54THEN120
 112 IFPEEK(18445)=1
THEN70
 113 GOTO90
 114 Z=1:POKE17677,0
 115 IFPEEK(18434)=0
THENIFPEEK(18433)=1T
HEN117
 116 IFPEEK(18436)=0
THEN1630
 117 R=16896:GOSUB11
8:R=17152:GOSUB118:G
OTO1610
 118 FORX=1TOM1+1:PO
KER,63
 119 R=R+1:NEXT:RETU
RN
 120 Y=0:T=0
 130 PRINT" ":PRINT"
DISPLAY ROUTINE":GOS
UB1180
 140 PRINT" ":PRINT"
PUSH FUNCTION BUTTON
":GOSUB1180
 150 PRINT" ":PRINT"
& THEN ENTER BUTTON"
:GOSUB1180
 160 IFPEEK(37633)=2
54THENT=0:GOTO190
 170 IFT=2000THEN122
0
 180 T=T+1:GOTO130
 190 FORY=1TO5:PRINT
" "
 200 IFY=1THENPRINT"
DISPLACEMENT OF RACK
":GOTO250
 210 IFY=2THENPRINT"
PRESENT ROLL":GOTO25
0
 220 IFY=3THENPRINT"
ROLL OF RACK SECTION
":GOTO250
 230 IFY=4THENPRINT"
PRESENT DISTANCE":GO
TO250
 240 PRINT"EXIT FROM
ROUTINE"
 250 GOSUB1180:IFPEE
K(37633)=254THENNEXT
Y:GOTO190
 260 IFPEEK(37633)=2
51THEN290
 270 IFT=500THEN120
 280 T=T+1:GOTO260
 290 IFY=2THEN990
 300 IFY=4THEN1060
 310 IFY=5THEN1220
 312 IFZ=1THEN320
 315 PRINT" ":PRINT"
THIS IS FIRST RUN"
 317 GOSUB1190:GOTO1
20
 320 T=0:N=0
 330 PRINT" ":PRINT"
PUSH STEP FOR RACK #
"
 340 IFPEEK(37633)=2
53THEN365
 350 IFT=500THEN120
 360 T=T+1:GOTO340
 365 GOSUB370:GOSUB4
55:GOTO650
 370 PRINT" ":PRINT"
USE LEFT/RIGHT SW.":
GOSUB1190
 380 PRINT" ":PRINT"
TO SELECT DIGIT":GOS
UB1190
 390 PRINT" ":PRINT"
PUSH ENTER WHEN OK":
GOSUB1190
 400 Y1=0:Y2=Y1:Y3=Y
2:PRINT" "
 404 IFN=0THENPRINTY
1;Y2;Y3:GOTO410
 406 IFR=0THENPRINTY
1;Y2;Y3"< LEFT END":
GOTO410
 408 PRINTY1;Y2;Y3"<
RIGHT END"
 410 P=1
 420 IFP=2THENPOKE39
682,2:GOTO450
 430 IFP=3THENPOKE39
682,4:GOTO450
 440 POKE39682,1
 450 GOSUB1190:RETUR
N
 455 IFPEEK(37633)=2
39THENP=P+1:GOTO500
 460 IFPEEK(37633)<>
247THEN520
 470 P=P-1
 480 IFP=0THENP=3
 490 GOTO420
 500 IFP=4THEN410
 510 GOTO420
 520 IFPEEK(37633)<>
253THENRETURN
 530 FORY=0TO9:PRINT
" "
 540 IFP=2THENY2=Y:G
OTO570
 550 IFP=3THENY3=Y:G
OTO570
 560 Y1=Y
 570 IFN=0THENPRINTY
1;Y2;Y3:GOTO600
 580 IFR=0THENPRINTY
1;Y2;Y3"< LEFT END":
GOTO600
 590 PRINTY1;Y2;Y3"<
RIGHT END"
 600 GOSUB1180
 610 IFPEEK(37633)=2
53THENNEXT:GOTO530
 620 RETURN
```

Figure A.4-2 BASIC Program Listing

```
650 IFPEEK(37633)=2
51THEN680
660 IFT=100000THENP
OKE39682,0:GOTO320
670 T=T+1:GOSUB455:
GOTO650
680 POKE39682,0:IFY
1=0THENIFY2=0THENIFY
3=0THEN320
690 X=Y1*100+Y2*10+
Y3:IFX<=M1+1THEN710
700 PRINT" ":PRINT"
ENTRY TOO LARGE":GOS
UB1190:GOTO320
710 T1=0:PRINT" "
720 PRINT"USE UP/DO
WN SWITCH":GOSUB1190
:PRINT" "
730 PRINT"TO CHANGE
RACK NO.":GOSUB1190
740 GOSUB1160
760 Y1=9:IFY=3THEN8
20
770 IFE(X)<0THENY1=
5:E(X)=-E(X)
780 M=INT(E(X)*3.28
+K3)/100
790 IFY1<>5THENPRIN
T" ":PRINT"RACK"X"F"
M"FEET":GOTO870
800 PRINT" ":PRINT"
RACK"X"G"M"FEET":E(X
)=-E(X):GOTO870
820 M=X*2+16382:Y2=
PEEK(M):Y3=PEEK(M+1)
830 IFY3>=8THENY2=2
56-Y2:Y3=15-Y3:Y1=5
840 Y2=INT((Y3*256+
Y2)*.078125+K3)/10
850 IFY1=5THENPRINT
" ":PRINT"RACK"X"G"Y
2"DEG":GOTO870
860 PRINT" ":PRINT"
RACK"X"F"Y2"DEG"
870 GOSUB1180
880 IFPEEK(37633)=2
47THEN920
890 IFPEEK(37633)<>
239THEN940
900 X=X-1:IFX=0THEN
X=X+1:GOTO940
910 GOTO760
920 X=X+1:IFX<=M1+1
THEN760
930 X=X-1
940 IFPEEK(37633)=2
51THEN1220
950 IFPEEK(37633)=2
54THEN120
960 IFT<>500THENT=T
+1:GOTO870
970 IFT1=4THEN120
980 T1=T1+1:GOTO720
990 GOSUB1160
992 FORN=1TO2:Y1=9
994 Y2=PEEK(40192)A
ND224
998 Y3=PEEK(40194)A
ND15
1000 IFY3>=8THENY2=
256-Y2:Y3=15-Y3:Y1=5
1010 S(N)=Y3*256+Y2
:NEXT
1014 IFS(1)<>S(2)TH
EN992
1016 Y2=INT(S(1)*.0
78125+K3)/10
1030 IFY1=5THENPRIN
T" ":PRINT"ROLL G"Y2
"DEGREES":GOTO1100
1040 PRINT" ":PRINT
"ROLL F"Y2"DEGREES":
GOTO1100
1060 GOSUB1160
1061 FORN=1TO2:Y2=P
EEK(40704)
1062 Y3=PEEK(40706)
1064 S(N)=Y3*256+Y2
:NEXT
1068 IFS(1)<>S(2)TH
EN1061
1070 Y2=INT(S(1)*1.
322835+K3)/10
1090 PRINT" ":PRINT
"LOCATION"Y2"FEET"
1100 GOSUB1190:IFPE
EK(37633)=251THEN122
0
1110 IFPEEK(37633)=
254THEN120
1130 IFT=2000THEN12
0
1140 T=T+1:IFY=2THE
N992
1150 GOTO1061
1160 PRINT" ":PRINT
"PUSH EXIT TO EXIT":
GOSUB1190:PRINT" "
1170 PRINT"& FUNCTI
ON TO RETURN":GOSUB1
190:T=0:RETURN
1180 FORD=1TO1000:N
EXTD:RETURN
1190 FORD=1TO3000:N
EXTD:RETURN
1220 IFZ=1THEN1610
1230 GOTO70
1240 PRINT" ":PRINT
"PROCESSING"
1290 N1=M1-1:N2=2*N
1:K=75.6:K1=.0000958
557
1300 BP=12288:BI=12
800:AP=13312:AI=1382
4:AM=14336
1310 N=1:FORU=0TON2
STEP2
1320 BL=PEEK(BP+U):
BH=PEEK(BP+U+1)
1325 IFBH>127THENBL
=BL-256:BH=BH-255
1330 IL=PEEK(BI+U):
IH=PEEK(BI+U+1)
1335 IFIH>127THENIL
=IL-256:IH=IH-255
1340 AL=PEEK(AP+U):
AH=PEEK(AP+U+1)
1345 IFAH>127THENAL
=AL-256:AH=AH-255
1350 SL=PEEK(AI+U):
SH=PEEK(AI+U+1)
1355 IFSH>127THENSL
=SL-256:SH=SH-255
1360 ML=PEEK(AM+U):
MH=PEEK(AM+U+1)
1365 IFMH>127THENML
=ML-256:MH=MH-255
1370 D(N)=BL-IL+(BH
-IH)*256
1380 C(N)=AL-SL+ML+
(AH-SH+MH)*256
1390 N=N+1:NEXT
1400 FORN=1TON1:D(N
+1)=D(N)+D(N+1):NEXT
1410 B=(D(M1)/M1)*K
1:C(1)=C(1)*K1-B
1420 FORN=2TOM1:C(N
)=C(N)*K1-B+C(N-1):N
EXT
1430 Y=0:FORN=1TOM1
:Y=Y+SIN(C(N)):NEXT
```

Figure A.4-3 BASIC Program Listing continued

```
 1440 POKE188,61:POK
E189,214:Y=(Y1-Y2/K2
)/K-Y
 1450 V=Y/(M1+1):R=A
TN(V/SQR(-V*V+1))
 1460 D(1)=K*SIN(R)
 1470 FORN=1TOM1:D(N
+1)=K*SIN(C(N)+R)+D(
N)
 1480 NEXT
 1490 FORN=1TOM1+1
 1520 E(N)=D(N):IFD(
N)>0THEN1560
 1530 IFD(N)=0THEN15
80
 1540 C(N)=INT(-D(N)
*K2+K3)+48:IFC(N)=48
THEN1580
 1550 D(N)=45:GOTO15
90
 1560 D(N)=INT(D(N)*
K2+K3)+48:IFD(N)=48T
HEN1580
 1570 C(N)=45:GOTO15
90
 1580 D(N)=48:C(N)=4
8
 1590 POKE16895+N,D(
N):POKE17151+N,C(N)
 1600 NEXT
 1610 POKE4,0:POKE5,
210:N=USR(0)
 1620 GOTO90
 1630 PRINT" ":PRINT
"END-POINT ROUTINE":
GOSUB1180
 1640 PRINT" ":PRINT
"PUSH FUNCTION BUTTO
N":GOSUB1180
 1650 PRINT" ":PRINT
"TO BEGIN":GOSUB1180
 1660 R=0:IFPEEK(376
33)<>254THEN1630
 1670 T=0:N=1
 1680 PRINT" ":PRINT
"PUSH STEP TO ENTER"
:GOSUB1180
 1690 PRINT" ":PRINT
"END-POINT LOCATION"
:GOSUB1180
 1700 IFPEEK(37633)=
253THEN1730
 1710 IFT=500THEN163
0
 1720 T=T+1:GOTO1680
 1730 GOSUB370
 1740 GOSUB455
 1750 IFPEEK(37633)=
251THEN1780
 1760 IFT=500THENPOK
E39682,0:GOTO1670
 1770 T=T+1:GOTO1740
 1780 POKE39682,0:IF
R=1THEN1800
 1790 X=Y1*100+Y2*10
+Y3:R=1:GOTO1670
 1800 Y2=Y1*100+Y2*1
0+Y3:Y1=X/K2:GOTO124
0
 1810 END
```

Figure A.4-4 BASIC Program Listing continued

A.5 Normalization Program

Whenever the Register Setup program is executed the coal face measurement system is placed on a ready status. In this routine the system power is checked, all of the temporary control registers are setup, and if necessary the tape is rewound.

Figure A.5-1 shows the flowchart while Figure A.5-2 through A.5-6 presents the actual program listing.

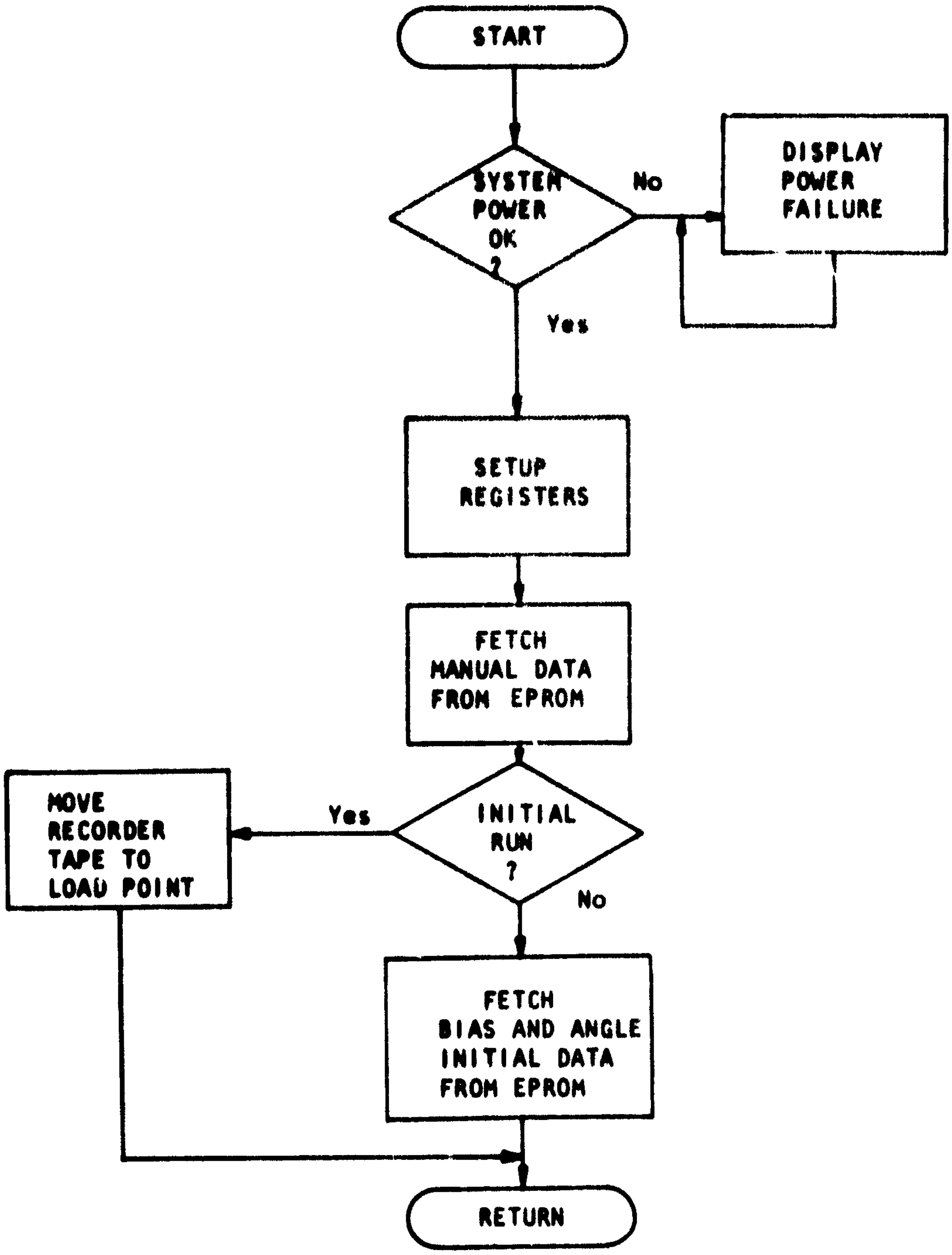


Figure A.5-1 Register Setup Program Flowchart

```
*=$6400
----------------------
NORMALIZE ROUTINE

==6400 LOC=$00FE

==6400 LOCA=$4508

==6400 NUMANG=$4620

==6400 RUNLOW=$4621

==6400 RUNHI=$4622

==6400 RML=$4801

==6400 RMH=$4802

==6400 CODE=$4515

==6400 RMLHIB=$4516

==6400 HCOUNT=$450E

==6400 LCOUNT=$450F

==6400 PPROM=$D800

==6400 RPROM=$DA00

==6400 PROMH=$D100

==6400 PROML=$D150

==6400 FUNC=$9301

==6400 PLIER=$4517

;PORTS

==6400 PAC=$9500

==6400 PBC=$9502

==6400 PBD=$9602
```

```
;SAVE ZERO CONSTANT

A0FF   LDY #$FF
==6402 L
C8     INY
B90000 LDA $0000,Y
48     PHA
C012   CPY #$12
D0F7   BNE L

;CLEAR DISPLAY

20F0E9 JSR $E9F0 CRL
F

;NORMALIZING

A000   LDY #$00
==6410 LOADNO
B91E64 LDA MSGNOR,Y
C93B   CMP #';'
F013   BEQ ALT
2005EF JSR $EF05 OUT
D
C8     INY
4C1064 JMP LOADNO

==641E MSGNOR
4E4F   .BYT 'NORMALI
ZING;'

;ALTER NMI VECTOR

==642A ALT
A900   LDA #$00
8D02A4 STA $A402
A968   LDA #$68
8D03A4 STA $A403

;SYSTEM POWER STATUS

AD0296 LDA PBD
3003   BMI AAA
4CD866 JMP BATERR

==643C
;TURN RECORDER ON

==643C AAA
A908   LDA #$08
8D0296 STA PBD
```

```
;TURN-ON DELAY

20AA66 JSR DELAY1

;CASSETTE LOADED ?

AD0095 LDA PAC
0A     ASL A
1028   BPL CCC

==644A BBB
20F0E9 JSR $E9F0 CRL
F
A000   LDY #$00
==644F LOADCL
B95D64 LDA MSGCLE,Y
C93B   CMP #';'
F0F4   BEQ BBB
2005EF JSR $EF05 OUT
D
C8     INY
4C4F64 JMP LOADCL

==645D MSGCLE
4341   .BYT 'CASSETT
E LOADING ERR;'

==6472
;TAPE POWER STATUS

==6472 CCC
AD0095 LDA PAC
1003   BPL DDD
4CD866 JMP BATERR

;TURN RECORDER OFF

==647A DDD
A900   LDA #$00
8D0296 STA PBD

;TURN-OFF DELAY

209566 JSR DELAY
```

Figure A.5-2 Register Setup Program Listing

```
;ZERO-PAGE CONSTANTS

A900    LDA #$00 ID
85F2    STA $F2
A946    LDA #$46
85F3    STA $F3

==648A
A900    LDA #$00 BI
85F4    STA $F4
A932    LDA #$32
85F5    STA $F5

A900    LDA #$00 AI
85F6    STA $F6
A936    LDA #$36
85F7    STA $F7

==649A
A900    LDA #$00 BP
85F8    STA $F8
A930    LDA #$30
85F9    STA $F9

A900    LDA #$00 ROLL
85FA    STA $FA
A940    LDA #$40
85FB    STA $FB

==64AA
A900    LDA #$00 AP
85FC    STA $FC
A934    LDA #$34
85FD    STA $FD

;TEMP. RAM CONSTANTS

A904    LDA #$04 AIPR
8D0745  STA $4507

A900    LDA #$00
8D0A45  STA $450A RTY
P

==64BC
8D0B45  STA $450B PNT
R

8D0C45  STA $450C CNT
R

8D0D45  STA $450D DRF

8D1445  STA $4514 ZFR

8D0448  STA $4804 RAE

8D0748  STA $4807 CTE

==64CE
8D0A48  STA $480A TFR

A951    LDA #$51 YR
8D0346  STA $4603

;GET RUN # PAGE CODE

A915    LDA #$15
8500    STA $00
A945    LDA #$45
8501    STA $01

==64DE
A917    LDA #$17
8502    STA $02
A902    LDA #$02
8503    STA $03

A916    LDA #$16
8504    STA $04
A945    LDA #$45
8505    STA $05

==64EE
2000DA  JSR RPROM

;GET NO. OF ANGLES

A920    LDA #$20
8500    STA $00
A946    LDA #$46
8501    STA $01

A91C    LDA #$1C
8502    STA $02
A902    LDA #$02
==64FF
8503    STA $03

A921    LDA #$21
8504    STA $04
A946    LDA #$46
8505    STA $05

2000DA  JSR RPROM

;MANUAL DATA START

A900    LDA #$00
8500    STA $00
==6510
8D0845  STA LOCA
A938    LDA #$38
8501    STA $01
8D0945  STA LOCA+1

;MANUAL DATA END

20BF66  JSR LIMIT

;MANUAL DATA EPROM

A91E    LDA #$1E
8502    STA $02
==6521
A902    LDA #$02
8503    STA $03

2000DA  JSR RPROM

;INITIAL KEY TURNED?

AD0193  LDA FUNC
C97F    CMP #$7F
F003    BEQ YES
4C0166  JMP RUN

==6532
;TURN RECORDER ON

==6532  YES
A908    LDA #$03
8D0296  STA PBD

;TURN-ON DELAY

20AA66  JSR DELAY1

;TAPE TO LOAD POINT
;LOAD COMMAND READY

A977    LDA #$77
8D0295  STA PBC

;INTERMEDIATE DELAY

209566  JSR DELAY
```

Figure A.5-3 Register Setup Program Listing continued

```
==6542
;LOAD COMMAND SENT
;STABILIZE RECORDER

A957   LDA #$57
8D0295 STA PBC

;LOAD POINT LOOP

==6547 LPLOOP
AD0095 LDA PAC
2910   AND #$10
D0F9   BNE LPLOOP

;TURN-OFF DELAY

209566 JSR DELAY

;TURN RECORDER OFF

A900   LDA #$00
8D0296 STA PBD

;SET RUN NUMBER TO 0
;READ RMH

A908   LDA #$08
==6558
8D0E45 STA HCOUNT

==655B SHREAD
2000D1 JSR PROMH
2000DA JSR RPROM

;EPROM ADDR. EMPTY ?

AD0248 LDA RMH
C9FF   CMP #$FF
F009   BEQ HZERO

;LOOK AT NEXT ADDR.

8D2246 STA RUNHI
==656B
EE0E45 INC HCOUNT
4C5B65 JMP SHREAD

;PREVIOUS ADDR. =0 ?

==6571 HZERO
AD2246 LDA RUNHI
D006   BNE HSTORE
8D0248 STA RMH
4C8865 JMP SLOW

;EPROM RMH ADDR TO 0

==657C HSTORE
EE0248 INC RMH
2000D1 JSR PROMH
2000D8 JSR PPROM
EE0E45 INC HCOUNT

;READ RML

==6588 SLOW
A900   LDA #$00
8D0F45 STA LCOUNT
8D1645 STA RMLHIB

==6590 SLREAD
2050D1 JSR PROML
2000DA JSR RPROM

;EPROM ADDR. EMPTY ?

AD0148 LDA RML
C9FF   CMP #$FF
F011   BEQ SSET

;LOOK AT NEXT ADDR.

==659D SINCR
AD0148 LDA RML
8D2146 STA RUNLOW
EE0F45 INC LCOUNT
D0E8   BNE SLREAD
EE1645 INC RMLHIB
4C9065 JMP SLREAD

==65AE
;NEAR END OF PAGE ?

==65AE SSET
AD2146 LDA RUNLOW
C9FE   CMP #$FE
D005   BNE LZERO

;PAGE CODE =0 ?

AD1545 LDA CODE
D0E3   BNE SINCR

;PREVIOUS ADDR. =0 ?

==65BA LZERO
AD2146 LDA RUNLOW
D006   BNE LSTORE
8D0148 STA RML
4CD665 JMP NORMKE

;EPROM RML ADDR TO 0

==65C5 LSTORE
EE0148 INC RML
2050D1 JSR PROML
2000D8 JSR PPROM
EE0F45 INC LCOUNT
D003   BNE NORMKE
EE1645 INC RMLHIB

==65D6
;NORMALIZE KEY

==65D6 NORMKE
20F0E9 JSR $E9F0 CRL
F
A000   LDY #$00

==65DB LOADNK
B9E965 LDA MSG1,Y
C93B   CMP #';'
F015   BEQ KEY
2005EF JSR $EF05 OUT
D
C8     INY
4CDB65 JMP LOADNK

==65E9 MSG1
4E4F    .BYT 'NORMALI
ZE KEY;'

;INITIAL KEY TURNED?

==65F7 KEY
AD0193 LDA FUNC
C97F   CMP #$7F
F0F9   BEQ KEY
4CF966 JMP RETURN

;GET RUN NUMBER
;READ RMH

==6601 RUN
A908   LDA #$08
8D0E45 STA HCOUNT

==6606 RHREAD
2000D1 JSR PROMH
2000DA JSR RPROM

;EPROM ADDR. EMPTY ?

AD0248 LDA RMH
C9FF   CMP #$FF
F009   BEQ RLOW
```

Figure A.5-4 Register Setup Program Listing continued

```
;LOOK AT NEXT ADDR.

8D2246 STA RUNHI
==6616
EE0E45 INC HCOUNT
4C0666 JMP RHREAD

;READ RML

==661C RLOW
A900   LDA #$00
8D0F45 STA LCOUNT
8D1645 STA RMLHIB

==6624 RLREAD
2050D1 JSR PROML
2000DA JSR RPROM

;EPROM ADDR. EMPTY ?

AD0148 LDA RML
C9FF   CMP #$FF
F011   BEQ RSET

;LOOK AT NEXT ADDR.

==6631 RINCR
AD0148 LDA RML
8D2146 STA RUNLOW
EE0F45 INC LCOUNT
D0E6   BNE RLREAD
EE1645 INC RMLHIB
4C2466 JMP RLREAD

==6642
;NEAR END OF PAGE ?

==6642 RSET
AD2146 LDA RUNLOW
C9FE   CMP #$FE
D005   BNE TRANS

;PAGE CODE =0 ?

AD1545 LDA CODE
F0E3   BEQ RINCR

;TRANSFER TO RMH/RML

==664E TRANS
AD2246 LDA RUNHI
8D0248 STA RMH
AD2146 LDA RUNLOW
8D0148 STA RML
```

```
;BIAS-INITIAL START

A900   LDA #$00
8500   STA $00
==665E
8D0845 STA LOCA
A932   LDA #$32
8501   STA $01
8D0945 STA LOCA+1

;BIAS-INITIAL END

20BF66 JSR LIMIT

;BIAS-INITIAL EPROM

A900   LDA #$00
8502   STA $02
==666F
A904   LDA #$04
8503   STA $03

2000DA JSR RPROM

;ANGLE-INITIAL START

A900   LDA #$00
8500   STA $00
8D0845 STA LOCA
A936   LDA #$36
==667F
8501   STA $01
8D0945 STA LOCA+1

;ANGLE-INITIAL END

20BF66 JSR LIMIT

;ANGLE-INITIAL EPROM

A900   LDA #$00
8502   STA $02
A906   LDA #$06
8503   STA $03

==668F
2000DA JSR RPROM
4CF966 JMP RETURN
```

```
;DELAY SUBROUTINE
;1 SECOND

==6695 DELAY
A903   LDA #$03
8D1745 STA PLIER
==669A DELA
A2FF   LDX #$FF
==669C NXT
A0FF   LDY #$FF
==669E NYT
88     DEY
D0FD   BNE NYT
CA     DEX
D0F8   BNE NXT
CE1745 DEC PLIER
D0F1   BNE DELA
60     RTS

;DELAY SUBROUTINE
;10 SECONDS

==66AA DELAY1
A91F   LDA #$1F
8D1745 STA PLIER
==66AF DELA10
A2FF   LDX #$FF
==66B1 NXXT
A0FF   LDY #$FF
==66B3 NYYT
88     DEY
D0FD   BNE NYYT
CA     DEX
D0F8   BNE NXXT
CE1745 DEC PLIER
D0F1   BNE DELA10
60     RTS
```

Figure A.5-5 Register Setup Program Listing continued

```
;DATA END POINT

==66BF LIMIT
A200   LDX #$00
AD2046 LDA NUMANG
0A     ASL A
9001   BCC LOW
E8     INX
==66C8 LOW
D8     CLD
18     CLC
6D0845 ADC LOCA
8504   STA $04
8A     TXA
D8     CLD
18     CLC
6D0945 ADC LOCA+1
8505   STA $05
60     RTS

==66D8
;BATTERY ERROR

==66D8 BATERR
20F0E9 JSR $E9F0 CRL
F
A000   LDY #$00
==66DD LOADBE
B9EB66 LDA MSGBER,Y
C93B   CMP #';'
F0F4   BEQ BATERR
2005EF JSR $EF05 OUT
D
C8     INY
4CDD66 JMP LOADBE

==66EB MSGBER
504F   .BYT 'POWER F
AILURE;'

;GET ZERO CONSTANTS

==66F9 RETURN
A012   LDY #$12
==66FB LP
68     PLA
990000 STA $0000,Y
88     DEY
10F9   BPL LP
60     RTS
```

Figure A.5-6 Register Setup Program Listing continued

A.6 Input/Store Program

The program that is executed most often by the system is the one that inputs and stores all of the data. This routine occurs each time the angle and distance electronics hardware sends an interrupt signal to the computer. When activated, the input resolver angle sum is divided by the number of samples and the result is temporarily stored for later calculations. The program is also designed to detect resolver failures which would not be averaged out by data reduction.

Figure A.6-1 shows the flowchart while Figure A.6-2 through A.6-5 presents the actual program listing.

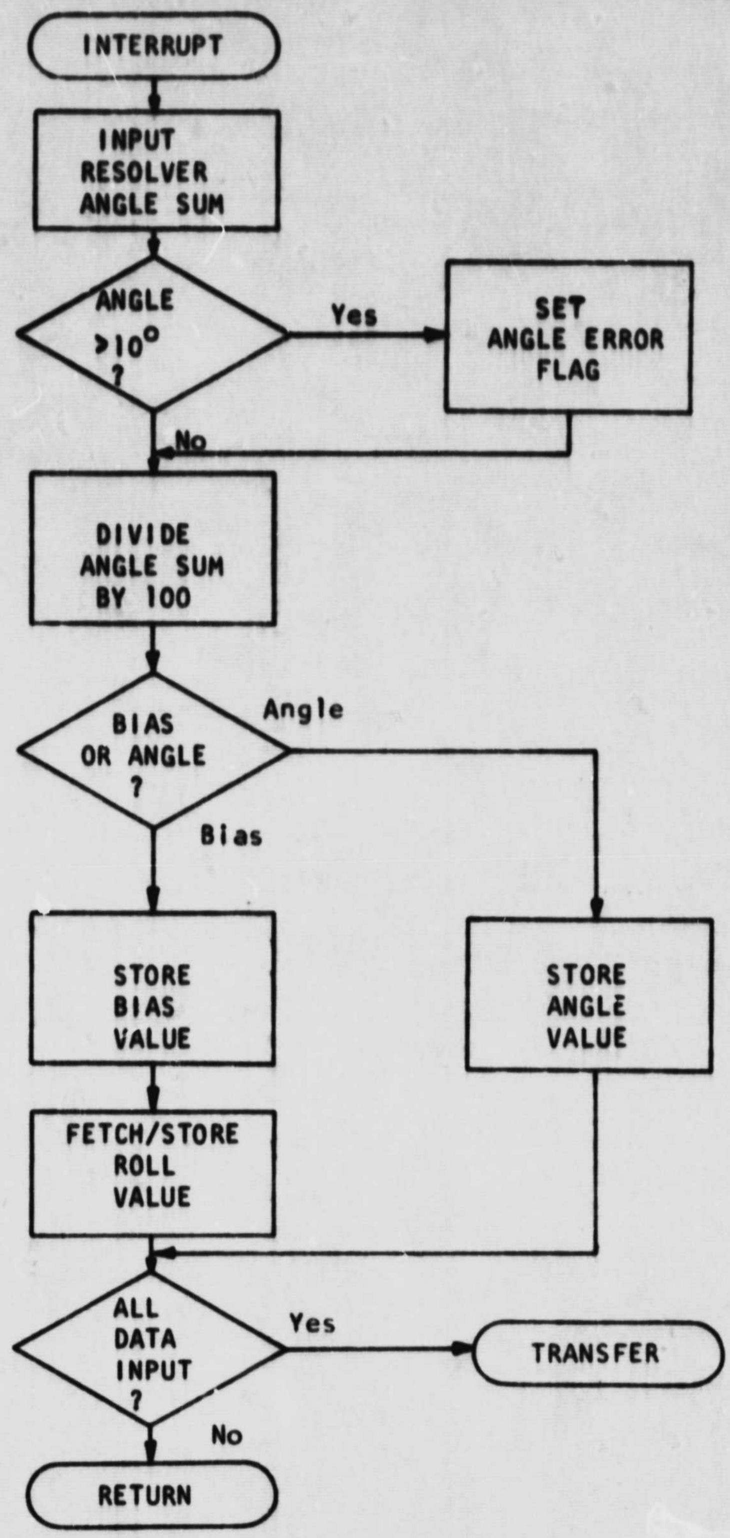


Figure A.6-1 Input/Store Program Flowchart

```
        *=$00F4
-------------------
INPUT/STORE ROUTINE


==00F4 BLOCI
       *=*+2
==00F6 ALOCI
       *=*+2
==00F8 BLOC
       *=*+2
==00FA RLOC
       *=*+2
==00FC ALOC
       *=*+2
;TEMP. RAM REGISTERS

==00FE
       *=$4500
==4500 RESULT
       *=*+1
==4501 LOWBYT
       *=*+1
==4502 RESLOW
       *=*+1
==4503 RESHI
       *=*+1
==4504 HIBYTE
       *=*+1
==4505 REMAIN
       *=*+1
==4506
       *=$450A
==450A RTYPE
       *=*+1
==450B POINTR
       *=*+1
==450C COUNTR
       *=*+1
==450D
       *=$4801
==4801 RML
       *=*+1
==4802 RMH
       *=*+1
==4803
       *=$4804
==4804 RAE
       *=*+1
;PROGRAM CONSTANTS
==4805
       *=$6800
==6800 PBD=$9602
```

```
==6800 PAL=$9D00

==6800 PBL=$9D02

==6800 PAR=$9E01

==6800 PBR=$9E00

==6800 IERDST=$9E0E

==6800 NUMANG=$4620


;PUSH A,X,Y TO STACK

48      PHA
8A      TXA
48      PHA
98      TYA
48      PHA

;SAVE ZERO CONSTANT

A0FF    LDY #$FF
==6807 L
C8      INY
B90000 LDA $0000,Y
48      PHA
C012    CPY #$12
D0F7    BNE L

;LOAD/SAVE RESOLVER

AD019E LDA PAR
8D0145 STA LOWBYT
AD009E LDA PBR
==6819
8D0445 STA HIBYTE
48      PHA

;CHANGE TO 2'S COMP.

1015    BPL LESS
49FF    EOR #$FF
AA      TAX
AD0145 LDA LOWBYT
49FF    EOR #$FF
8D0145 STA LOWBYT
==682A
EE0145 INC LOWBYT
D001    BNE NOCARY
E8      INX
==6830 NOCARY
8A      TXA
8D0445 STA HIBYTE
```

```
;SET REGISTER X TO 0

==6834 LESS
A200    LDX #$00

;RESOLVER ANGLE ERR

C92C    CMP #$2C
900C    BCC HUNDRD
AD0145 LDA LOWBYT
C970    CMP #$70
9005    BCC HUNDRD
A901    LDA #$01
8D0448 STA RAE

==6846
;HIGH BYTE <100 ?

==6846 HUNDRD
AD0445 LDA HIBYTE
C964    CMP #$64
9011    BCC DIVIDE

;SET FLAG GREATER

A2FF    LDX #$FF

;SUBTRACT 25,000

D8      CLD
18      CLC
AD0145 LDA LOWBYT
6958    ADC #$58
==6856
8D0145 STA LOWBYT
AD0445 LDA HIBYTE
699E    ADC #$9E
```

Figure A.6-2 Input/Store Program Listing

```
;DIVISION ROUTINE

==685E DIVIDE
A008   LDY #$08
38     SEC
E964   SBC #$64
==6863 DLOOP
08     PHP
2E0045 ROL RESULT
0E0145 ASL LOWBYT
2A     ROL A
28     PLP
9005   BCC ADD
E964   SBC #$64
4C7568 JMP NEXT
==6873 ADD
6964   ADC #$64
==6875 NEXT
88     DEY
D0EB   BNE DLOOP
B003   BCS LAST
6964   ADC #$64
18     CLC
==687D LAST
2E0045 ROL RESULT

;STRAIGHTEN OUT

8D0545 STA REMAIN
AD0045 LDA RESULT
8D0145 STA LOWBYT

;ADD 250 ?

E0FF   CPX #$FF
D00E   BNE SHIFT

==688D
;ADD 250

D8     CLD
18     CLC
AD0145 LDA LOWBYT
69FA   ADC #$FA
8D0145 STA LOWBYT
B002   BCS SHIFT
A200   LDX #$00
```

```
;NORMALIZE ANSWER

==689B SHIFT
4A     LSR A
4A     LSR A
4A     LSR A
4A     LSR A
E0FF   CPX #$FF
D002   BNE EIGHT
0910   ORA #$10
==68A5 EIGHT
8D0345 STA RESHI
AD0145 LDA LOWBYT
0A     ASL A
0A     ASL A
0A     ASL A
0A     ASL A
8D0245 STA RESLOW

;OBTAIN REMAINDER

AD0545 LDA REMAIN
==68B5
8D0445 STA HIBYTE
0A     ASL A
0A     ASL A
0A     ASL A
0A     ASL A
8D0145 STA LOWBYT
AD0445 LDA HIBYTE
4A     LSR A
4A     LSR A
4A     LSR A
==68C5
4A     LSR A
```

```
;DIVISION ROUTINE

A008   LDY #$08
38     SEC
E964   SBC #$64
==68CB LOOPD
08     PHP
2E0045 ROL RESULT
0E0145 ASL LOWBYT
2A     ROL A
28     PLP
9005   BCC AD
E964   SBC #$64
4CDD68 JMP NXT
==68DB AD
6964   ADC #$64
==68DD NXT
88     DEY
D0EB   BNE LOOPD
B003   BCS LST
6964   ADC #$64
18     CLC
==68E5 LST
2E0045 ROL RESULT

;ORIGINAL VALUE NEG?

68     PLA
3002   BMI NEG
A0FF   LDY #$FF

;ADJUST ANSWER

==68ED NEG
AD0245 LDA RESLOW
0D0045 ORA RESULT
8D0245 STA RESLOW

;CHANGE TO 2'S COMP.

C0FF   CPY #$FF
F015   BEQ POSITV
49FF   EOR #$FF
8D0245 STA RESLOW
==68FF
AD0345 LDA RESHI
49FF   EOR #$FF
8D0345 STA RESHI
EE0245 INC RESLOW
D003   BNE POSITV
EE0345 INC RESHI
```

Figure A.6-3 Input/Store Program Listing continued

```
==690F
;FETCH DATA POINTER

==690F POSITV
AC0B45 LDY POINTR

;BIAS(0) OR ANGLE(1)

AD0A45 LDA RTYPE
C9FF   CMP #$FF
F052   BEQ ANGLE

;FROM BIAS TO ANGLE

49FF   EOR #$FF
8D0A45 STA RTYPE

;INITIAL RUN ?

AD0248 LDA RMH
==6921
C900   CMP #$00
D007   BNE BIAS
AD0148 LDA RML
C900   CMP #$00
F030   BEQ BIASI

;LOAD/SAVE BIAS DATA

==692C BIAS
AD0245 LDA RESLOW
91F8   STA (BLOC),Y
C8     INY
AD0345 LDA RESHI
91F8   STA (BLOC),Y
88     DEY

;ROLL DELAY/BAT STAT

98     TYA
48     PHA
A2A3   LDX #$A3
==693C LXL
A0FF   LDY #$FF
==693E LYL
AD0296 LDA PBD
3003   BMI DDD
4CD569 JMP BATERR
==6946 DDD
88     DEY
D0F5   BNE LYL
CA     DEX
D0F0   BNE LXL
68     PLA
A8     TAY

;LOAD/SAVE ROLL DATA

AD009D LDA PAL
91FA   STA (RLOC),Y
C8     INY
AD029D LDA PBL
==6957
91FA   STA (RLOC),Y
4CC669 JMP EXIT

;LOAD/SAVE BIAS DATA

==695C BIASI
AD0245 LDA RESLOW
91F4   STA (BLOCI),Y
C8     INY
AD0345 LDA RESHI
91F4   STA (BLOCI),Y
88     DEY
4CC669 JMP EXIT

;FROM ANGLE TO BIAS

==696B ANGLE
49FF   EOR #$FF
8D0A45 STA RTYPE

;INITIAL RUN ?

AD0248 LDA RMH
C900   CMP #$00
D007   BNE ANGL
AD0148 LDA RML
C900   CMP #$00
==697C
F00E   BEQ ANGLI

;LOAD/SAVE ANGL DATA

==697E ANGL
AD0245 LDA RESLOW
91FC   STA (ALOC),Y
C8     INY
AD0345 LDA RESHI
91FC   STA (ALOC),Y
4C9769 JMP COUNT

;LOAD/SAVE ANGL DATA

==698C ANGLI
AD0245 LDA RESLOW
91F6   STA (ALOCI),Y
C8     INY
AD0345 LDA RESHI
91F6   STA (ALOCI),Y

;ALL ANGLES DONE ?

==6997 COUNT
AE0C45 LDX COUNTR
E8     INX
EC2046 CPX NUMANG
F056   BEQ YES

;STORE COUNTER

8E0C45 STX COUNTR

;INCREMENT POINTER

C8     INY

;STORE POINTER

8C0B45 STY POINTR

==69A7
;PAGE DONE ?

D01D   BNE EXIT

;INITIAL RUN ?

AD0248 LDA RMH
C900   CMP #$00
D010   BNE NOTIR
AD0148 LDA RML
C900   CMP #$00
D009   BNE NOTIR

==69B7
;SELECT NEXT PAGE

E6F5   INC $F5
E6FB   INC $FB
E6F7   INC $F7
4CC669 JMP EXIT

;SELECT NEXT PAGE

==69C0 NOTIR
E6F9   INC $F9
E6FB   INC $FB
E6FD   INC $FD
```

Figure A.6-4 Input/Store Program Listing continued

```
;GET ZERO CONSTANTS

==6906 EXIT
A012   LDY #$12
==6908 LP
68     PLA
990000 STA $0000,Y
88     DEY
10F9   BPL LP

;PULL Y,X,A FR STACK

68     PLA
A8     TAY
68     PLA
AA     TAX
68     PLA

;RETURN FRM INTERUPT

40     RTI

;SYSTEM POWER STATUS

==69D5 BATERR
20F0E9 JSR $E9F0 CRL
F
A000   LDY #$00
==69DA LOADBE
B9E869 LDA MSGBER,Y
C93B   CMP #';'
F0F4   BEQ BATERR
2005EF JSR $EF05 OUT
D
C8     INY
4CDA69 JMP LOADBE

==69E8 MSGBER
504F   .BYT 'POWER F
AILURE;'
```

```
;DISABLE NMI

==69F6 YES
A902   LDA #$02
8D0E9E STA IERDST

;RESET POINTR,COUNTR

A900   LDA #$00
8D0B45 STA POINTR
8D0C45 STA COUNTR

;GO TO DATA TRANSFER

4C006B JMP $6B00
```

Figure A.6-5 Input/Store Program Listing continued

A.7 Data Transfer Program

Each time the coal face is completely cut the Data Transfer Program is executed. If after cutting the coal the system power is still good then the accumulated data is copied to tape. If the system happens to be making an 'initial run' then the generated information is also retained on an EPROM.

Figure A.7-1 shows the flowchart while Figure A.7-2 through A.7-6 presents the actual program listing.

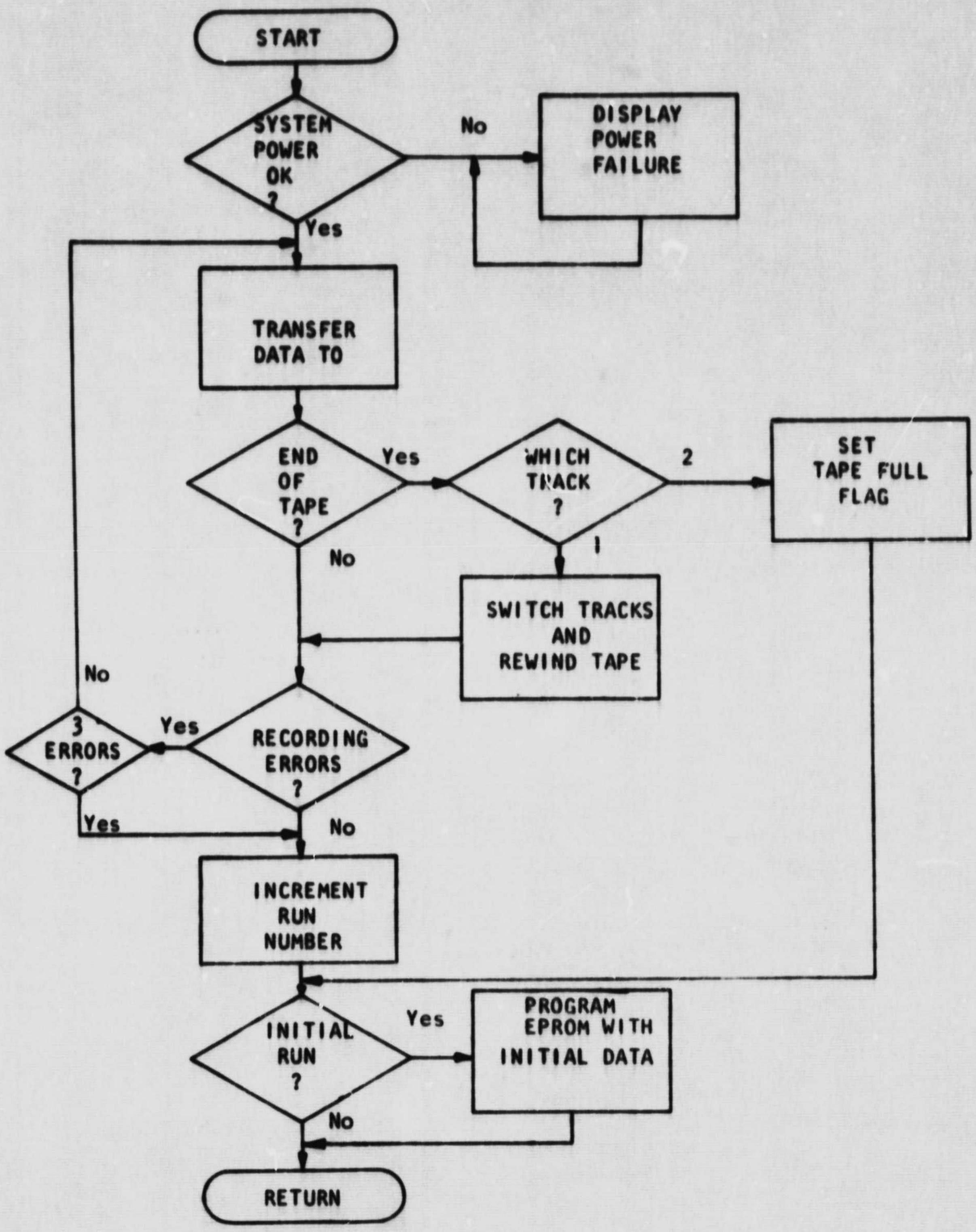


Figure A.7-1 Data Transfer Program Flowchart

```
      *=$00F2
-------------------
DUMP ROUTINE

==00F2 IDLOC
      *=*+2
==00F4 BLOCI
      *=*+2
==00F6 ALOCI
      *=*+2
==00F8 BLOC
      *=*+2
==00FA RLOC
      *=*+2
==00FC ALOC
      *=*+2
==00FE LOC
      *=*+2
;TEMP  RAM REGISTERS

==0100
      *=$4506
==4506 ERR
      *=*+1
==4507 AIPROM
      *=*+1
==4508 LOCA
      *=*+2
==450A
      *=$450D
==450D DRF
      *=*+1
==450E
      *=$4517
==4517 FLIER
      *=*+1
==4518
      *=$4800
==4800 PTR
      *=*+1
==4801 RML
      *=*+1
==4802 RMH
      *=*+1
==4803 DAY
      *=*+1
==4804
      *=$4807
==4807 CTE
      *=*+1
==4808
      *=$480A
==480A TFR
      *=*+1
```

```
;PROGRAM CONSTANTS
;INTERUPT PORTS
==480B
      *=$6B00
==6B00 PAC=$9500

==6B00 PBC=$9502

==6B00 PAD=$9600

==6B00 PBD=$9602

==6B00 IERDST=$9E0E

;OTHER

==6B00 CODE=$4515

==6B00 PROMH=$D100

==6B00 PROML=$D150

==6B00 FPROM=$D800

==6B00 RPROM=$DA00

==6B00 NUMANG=$4620

;TAPE FULL FLAG SET?

AD0A48 LDA TFR
C901   CMP #$01
D003   BNE CONT
4C306C JMP INCR

;DISPLAY TRANSFERING

==6B0A CONT
20F0E9 JSR $E9F0 CRL
F
A200   LDX #$00
==6B0F LOADW
BD1D6B LDA WORD,X
C93B   CMP #';'
F014   BEQ DUMP
2005EF JSR $EF05 OUT
D
E8     INX
4C0F6B JMP LOADW

==6B1D WORD
5452   .BYT 'TRANSFE
RRING;'
```

```
;IDENTIFICATION REG.

==6B2A DUMP
A000   LDY #$00
AD0148 LDA RML
91F2   STA (IDLOC),Y
C8     INY
AD0248 LDA RMH
91F2   STA (IDLOC),Y
C8     INY
AD0348 LDA DAY
==6B3B
91F2   STA (IDLOC),Y

;CLEAR ERROR COUNTER

A900   LDA #$00
8D0645 STA ERR

;FETCH POWER/TRACK

20AE6D JSR PWRTRK

2000DA JSR RPROM

;RECORDER ON/TRK SEL

AD0048 LDA PTR
==6B4B
8D0296 STA PBD

;TURN-ON DELAY

20456D JSR DELAY1

;TAPE POWER STATUS

AD0095 LDA PAC
1003   BPL AAA
4C8D6D JMP BATERR

;WRITE COMMAND SET
;WRITE ON
;DATA ENABLE LOW

==6B59 AAA
A987   LDA #$87
8D0295 STA PBC

;HEAD DELAY

20306D JSR DELAY
```

Figure A.7-2 Data Transfer Program Listing

```
;START COMMAND SET
;WRITE-START ON
;WRITE-FILE-MARK ON

==6B61 STRT
A980   LDA #$80
8D0295 STA PBC

;WRITE-END ON LOOP

==6B66 WELOOP
AD0095 LDA PAC
2901   AND #$01
D0F9   BNE WELOOP

;WRITE-END OFF LOOP

==6B6D WFLOOP
AD0095 LDA PAC
2901   AND #$01
F0F9   BEQ WFLOOP

;WRITE-START OFF
;WRITE-FILE-MARK OFF
;DATA ENABLE HIGH

A996   LDA #$96
8D0295 STA PBC

;STOP/START DELAY
;72MSEC

A239   LDX #$39
==6B7B SSDEL1
A0FF   LDY #$FF
==6B7D YLOOP1
88     DEY
D0FD   BNE YLOOP1
CA     DEX
D0F8   BNE SSDEL1

;DATA ENABLE LOW

A986   LDA #$86
8D0295 STA PBC

;DATA ERROR ?

AD0095 LDA PAC
2904   AND #$04
==6B8D
D010   BNE CLEAR
AD0645 LDA ERR
C904   CMP #$04
B006   BCS JUMP
EE0645 INC ERR
4C616B JMP STRT

;CASSETTE TAPE ERROR

==6B9C JUMP
4C736C JMP ERROR

;CLEAR ERROR COUNTER

==6B9F CLEAR
A900   LDA #$00
8D0645 STA ERR

;DUMP PORTION
;TRANSFER ID REG.

A5F2   LDA IDLOC
85FE   STA $FE
A5F3   LDA IDLOC+1
85FF   STA $FF
A000   LDY #$00

;WRITE-START ON

A984   LDA #$84
==6BB0
8D0295 STA PBC

;DATA READY ON LOOP

==6BB3 DRLOOP
AD0095 LDA PAC
2902   AND #$02
F0F9   BEQ DRLOOP

;FETCH AND SEND DATA

==6BBA IDLOOP
B1FE   LDA (LOC),Y
8D0096 STA PAD

;DATA READY ROUTINE

205A6D JSR DFLOOP

;INCREMENT LOCATION

C8     INY

;ALL DONE ?

C004   CPY #$04
D0F3   BNE IDLOOP

;TRANSFER BIAS DATA

A5F8   LDA BLOC
85FE   STA $FE
==6BCB
A5F9   LDA BLOC+1
85FF   STA $FF
20696D JSR TRANSF

;TRANSFER ANGLE DATA

A5FC   LDA ALOC
85FE   STA $FE
A5FD   LDA ALOC+1
85FF   STA $FF
20696D JSR TRANSF

==6BDD
;SET-UP ROLL TRANSFR

CE2046 DEC NUMANG

;TRANSFER ROLL DATA

A5FA   LDA RLOC
85FE   STA $FE
A5FB   LDA RLOC+1
85FF   STA $FF
20696D JSR TRANSF

;TRANSFER LOW BYTE

C8     INY
B1FE   LDA (LOC),Y
==6BEE
8D0096 STA PAD

;DATA READY ROUTINE

205A6D JSR DFLOOP

;TRANSFER LAST BYTE

C8     INY
B1FE   LDA (LOC),Y
8D0096 STA PAD
```

Figure A.7-3 Data Transfer Program Listing continued

```
;END OF BLOCK ON

A98C   LDA #$8C
8D0295 STA PBC

==6BFF
;WRITE-END ON LOOP

==6BFF LOOPWE
AD0095 LDA PAC
2901   AND #$01
D0F9   BNE LOOPWE

;END OF BLOCK OFF

A984   LDA #$84
8D0295 STA PBC

;WRITE-END OFF LOOP

==6C0B LOOPWF
AD0095 LDA PAC
2901   AND #$01
F0F9   BEQ LOOPWF

;STOP/START DELAY
;42MSEC

A221   LDX #$21
==6C14 SSDEL
A0FF   LDY #$FF
==6C16 YLOOP
88     DEY
D0FD   BNE YLOOP
CA     DEX
D0F8   BNE SSDEL

;STOP COMMAND SET
;WRITE-START OFF
;DATA ENABLE HIGH

A997   LDA #$97
8D0295 STA PBC

;STOPPING DELAY

20306D JSR DELAY

;RESET NO. OF ANGLES

EE2046 INC NUMANG

==6C27
;END OF TAPE ?

AD0095 LDA PAC
2908   AND #$08
D031   BNE TRACK1

;TRACK ?

AD0048 LDA PTR
2901   AND #$01
D008   BNE SELECT

;TAPE FULL FLAG SET

A901   LDA #$01
==6C37
8D0A48 STA TFR
4CBE6C JMP OFF

;SELECT TRACK 2

==6C3D SELECT
A9F8   LDA #$F8
8D0296 STA PBD
8D0048 STA PTR

20AE6D JSR PWRTRK

2000D8 JSR PPROM

;TAPE TO LOAD POINT
;LOAD COMMAND SET

A9A7   LDA #$A7
==6C4D
8D0295 STA PBC

;INTERMEDIATE DELAY

20306D JSR DELAY

;LOAD COMMAND SENT
;STABILIZE RECORDER

A987   LDA #$87
8D0295 STA PBC

;LOAD POINT LOOP

==6C58 LPLOOP
AD0095 LDA PAC
2910   AND #$10
D0F9   BNE LPLOOP

;DATA ERROR ?

==6C5F TRACK1
AD0095 LDA PAC
2904   AND #$04
D015   BNE CLR
AD0645 LDA ERR
C904   CMP #$04
B006   BCS ERROR
EE0645 INC ERR
==6C70
4C596B JMP AAA

;CASSETTE TAPE ERROR

==6C73 ERROR
A901   LDA #$01
8D0748 STA CTE
4C806C JMP INCR

;CLEAR ERROR COUNTER

==6C7B CLR
A900   LDA #$00
8D0645 STA ERR

;INCREMENT RUN NO.

==6C80 INCR
EE0148 INC RML
D00C   BNE LW
EE0248 INC RMH

;PUT RMH ON EPROM

2000D1 JSR PROMH
2000D8 JSR PPROM
4CB86C JMP LOWER

==6C91
;RML AT END OF PAGE?

==6C91 LW
AD0148 LDA RML
C9FF   CMP #$FF
D020   BNE LOWER

;END OF PAGE CODE

A900   LDA #$00
8D1545 STA CODE
```

Figure A.7-4 Data Transfer Program Listing continued

```
;PUT CODE ON EPROM

A915   LDA #$15
8500   STA $00
==6CA1
A945   LDA #$45
8501   STA $01

A917   LDA #$17
8502   STA $02
A902   LDA #$02
8503   STA $03

A916   LDA #$16
8504   STA $04
==6CB1
A945   LDA #$45
8505   STA $05

2000D8 JSR PPROM

;PUT RML ON EPROM

==6CB8 LOWER
2050D1 JSR PROML
2000D8 JSR PPROM

;TURN-OFF DELAY

==6CBE OFF
20306D JSR DELAY

;STABILIZE RECORDER

A957   LDA #$57
8D0295 STA PBC

;TURN-OFF DELAY

20306D JSR DELAY

;TURN RECORDER OFF

A900   LDA #$00
8D0296 STA PBD

==6CCE
;RUN #=1 ?

AD0248 LDA RMH
C900   CMP #$00
D047   BNE DRFLAG
AD0148 LDA RML
C901   CMP #$01
D040   BNE DRFLAG

;BIAS-INITIAL START

A5F4   LDA BLOCI
==6CDE
8500   STA $00
8D0945 STA LOCA
A5F5   LDA BLOCI+1
8501   STA $01
8D0945 STA LOCA+1

;BIAS-INITIAL END

20C76D JSR LIMIT

;BIAS-INITIAL EPROM

A900   LDA #$00
==6CEF
8502   STA $02
AD0745 LDA AIPROM
8503   STA $03

;GO PROGRAM PROM

2000D8 JSR PPROM

;ANGLE-INITIAL START

A5F6   LDA ALOCI
8500   STA $00
8D0845 STA LOCA
==6D00
A5F7   LDA ALOCI+1
8501   STA $01
8D0945 STA LOCA+1

;ANGLE-INITIAL END

20C76D JSR LIMIT

;ANGLE-INITIAL EPROM
A900   LDA #$00
8502   STA $02
EE0745 INC AIPROM
==6D11
EE0745 INC AIPROM
AD0745 LDA AIPROM
8503   STA $03

;GO PROGRAM PROM

2000D8 JSR PPROM

;DATA REDUCTION SET

==6D1C DRFLAG
A9FF   LDA #$FF
8D0D45 STA DRF

;EXIT FROM ROUTINE

A012   LDY #$12
==6D23 LP
68     PLA
990000 STA $0000,Y
88     DEY
10F9   BPL LP

;PULL Y,X,A FR STACK

68     PLA
A8     TAY
68     PLA
AA     TAX
68     PLA

;RETURN FRM INTERUPT

40     RTI

;DELAY SUBROUTINE
;1 SECOND

==6D30 DELAY
A903   LDA #$03
8D1745 STA PLIER
==6D35 DELA
A2FF   LDX #$FF
==6D37 NXT
A0FF   LDY #$FF
==6D39 NYT
88     DEY
D0FD   BNE NYT
CA     DEX
D0F8   BNE NXT
CE1745 DEC PLIER
D0F1   BNE DELA
60     RTS
```

Figure A.7-5 Data Transfer Program Listing continued

```
;DELAY SUBROUTINE
;10 SECONDS

==6D45 DELAY1
A91F   LDA #$1F
8D1745 STA PLIER
==6D4A DELA1
A2FF   LDX #$FF
==6D4C NXXT
A0FF   LDY #$FF
==6D4E NYYT
88     DEY
D0FD   BNE NYYT
CA     DEX
D0F8   BNE NXXT
CE1745 DEC PLIER
D0F1   BNE DELA1
60     RTS

;DATA READY OFF LOOP

==6D5A DFLOOP
AD0095 LDA PAC
2902   AND #$02
D0F9   BNE DFLOOP

;DATA READY ON LOOP

==6D61 LOOPDR
AD0095 LDA PAC
2902     D #$02
F0F9   BEQ LOOPDR
60     RTS

;DATA TRANSFER SUB

==6D69 TRANSF
A200   LDX #$00
A000   LDY #$00

==6D6D DTLOOP
B1FE   LDA (LOC),Y
8D0096 STA PAD

;DATA READY ROUTINE

205A6D JSR DFLOOP

;INCREMENT LOCATION

C8     INY

B1FE   LDA (LOC),Y
8D0096 STA PAD

;DATA READY ROUTINE

205A6D JSR DFLOOP

==6D7E
;INCREMENT COUNT

E8     INX

;ALL DONE ?

EC2046 CPX NUMANG
F008   BEQ DONE

;INCREMENT LOCATION

C8     INY

;PAGE DONE ?

D0E6   BNE DTLOOP

;INCREMENT PAGE

E6FF   INC LOC+1
4C6D6D JMP DTLOOP

==6D8C DONE
60     RTS

;BATTERY ERROR

==6D8D BATERR
20F0E9 JSR $E9F0 CRL
F
A000   LDY #$00
==6D92 LOADBE
B9A06D LDA MSGBER,Y
C93B   CMP #';'
F0F4   BEQ BATERR
2005EF JSR $EF05 OUT
D
C8     INY
4C926D JMP LOADBE

==6DA0 MSGBER
504F   .BYT 'POWER F
AILURE;'

;SET-UP PWR/TRK REG.

==6DAE PWRTRK
A900   LDA #$00
8500   STA $00
A948   LDA #$48
8501   STA $01

A912   LDA #$12
8502   STA $02
A902   LDA #$02
8503   STA $03

==6DBE
A901   LDA #$01
8504   STA $04
A948   LDA #$48
8505   STA $05

60     RTS

;INITIAL DATA END

==6DC7 LIMIT
A200   LDX #$00
AD2046 LDA NUMANG
0A     ASL A
9001   BCC LOW
E8     INX
==6DD0 LOW
D8     CLD
18     CLC
6D0845 ADC LOCA
8504   STA $04
8A     TXA
D8     CLD
18     CLC
6D0945 ADC LOCA+1
8505   STA $05
60     RTS
```

Figure A.7-6 Data Transfer Program Listing continued

A.8 Contour Display Program

After the BASIC routine reduces the coordinate data to a useable form the results are presented to the operator by the Contour Display program. This program is designed to provide the displacement of the track to the nearest one-half foot in either the FACE or GOB direction. Additionally it displays any of the possible system errors which could theoretically occur.

Figure A.8-1 shows the flowchart while Figure A.8-2 through A.8-4 presents the actual program listing.

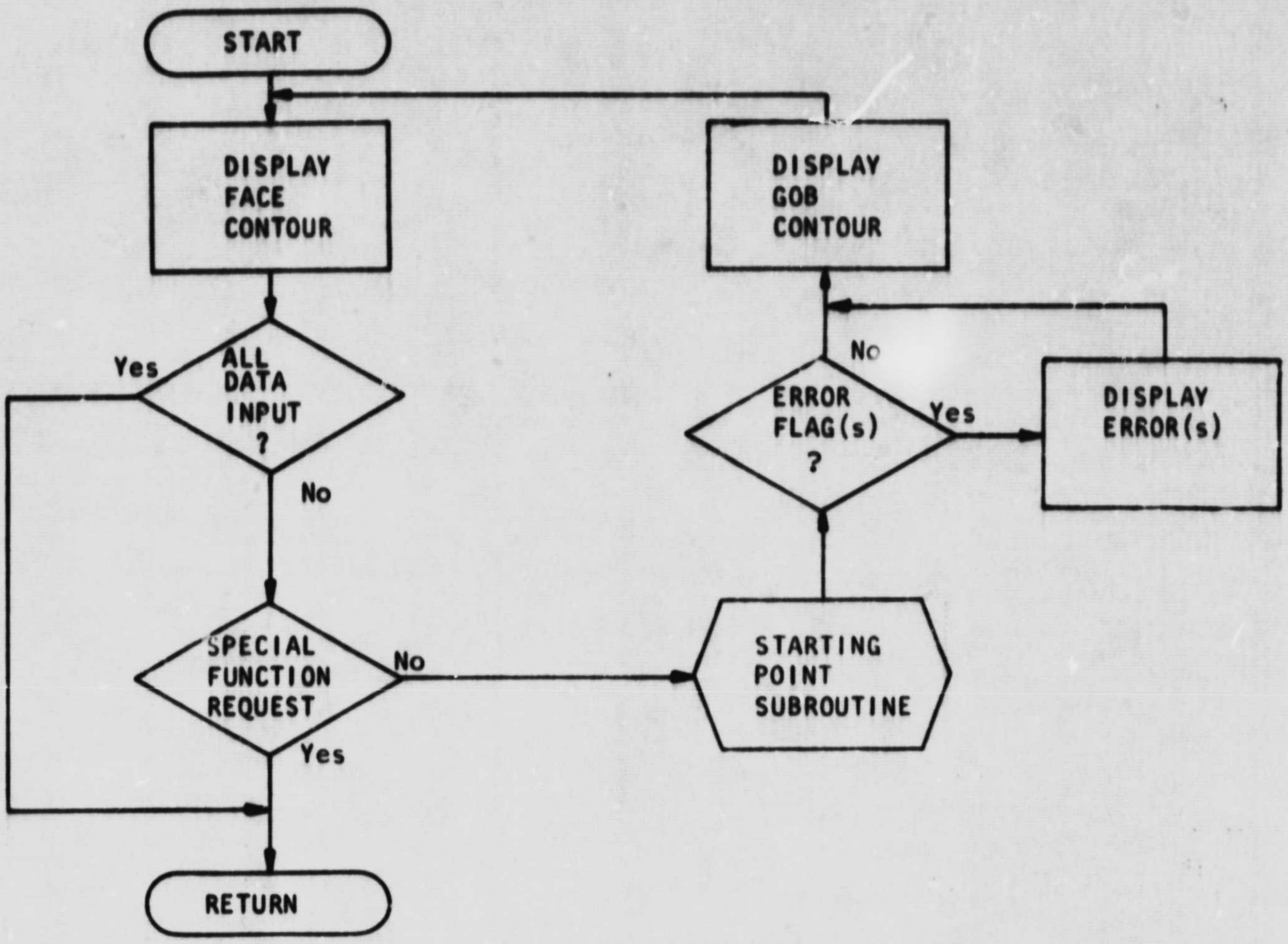


Figure A.8-1 Contour Display Program Flowchart

```
    *=$4510
---------------------
DISPLAY SUBROUTINE


==4510
;TEMP. RAM REGISTERS

==4510 ANSWER
       *=*+1
==4511 LOWMEM
       *=*+1
==4512 MPLIER
       *=*+1
==4513 LICAND
       *=*+1
==4514
       *=$8200
==8200
;PROGRAM CONSTANTS

==8200 LOC=$00FE

==8200 OUTD=$EF05

==8200 DRF=$450D

==8200 PLIER=$4517

==8200 NUMANG=$4620

==8200 RAE=$4804

==8200 CTE=$4807

==8200 TFR=$480A

==8200 FUNC=$9301

==8200 START=$D170

;SET-UP FACE DATA

==8200 FACE
A900   LDA #$00
85FE   STA $FE
A942   LDA #$42
85FF   STA $FF

;DISPLAY SUBROUTINE

204A82 JSR DISPLY

;DISPLAY FACE

A000   LDY #$00
==820D LOOPF
B91B82 LDA MSG1,Y
C93B   CMP #';'
F00D   BEQ SUB1
2005EF JSR OUTD
C8     INY
4C0D82 JMP LOOPF

==821B MSG1
2046   .BYT ' FACE;'

;JUMP TO SUBROUTINE

==8221 SUB1
20B882 JSR DELERR

;SET-UP GOB DATA

A900   LDA #$00
85FE   STA $FE
A943   LDA #$43
85FF   STA $FF

;DISPLAY SUBROUTINE

204A82 JSR DISPLY

;DISPLAY GOB

A000   LDY #$00
==8231 LOOPG
B93F82 LDA MSG2,Y
C93B   CMP #';'
F00C   BEQ SUB2
2005EF JSR OUTD
C8     INY
4C3182 JMP LOOPG

==823F MSG2
2047   .BYT ' GOB;'

;JUMP TO SUBROUTINE

==8244 SUB2
20B882 JSR DELERR

;GO BACK TO FACE

4C0082 JMP FACE

;DISPLAY SUBROUTINE

==824A DISPLY
20F0E9 JSR $E9F0 CRL
F

;DISPLAY FIRST ZERO

A930   LDA #$30
2005EF JSR OUTD

;MULTIPLICATION PART

A901   LDA #$01
8D1745 STA PLIER
8D1245 STA MPLIER
==825A
D8     CLD
==825B MULTIP
AE2046 LDX NUMANG
E8     INX
8E1345 STX LICAND
A900   LDA #$00
8D1145 STA LOWMEM
A208   LDX #$08
==8269 LOOPM
4E1245 LSR MPLIER
9004   BCC NOADD
18     CLC
6D1345 ADC LICAND
==8272 NOADD
6A     ROR A
6E1145 ROR LOWMEM
CA     DEX
D0F0   BNE LOOPM
```

Figure A.8-2 Contour Display Program Listing

```
;DIVISION ROUTINE

A008   LDY #$08
38     SEC
E90E   SBC #$0E
==827E LOOPDV
08     PHP
2E1045 ROL ANSWER
0E1145 ASL LOWMEM
2A     ROL A
28     PLP
9005   BCC ADDITN
E90E   SBC #$0E
4C9082 JMP NEXT1
==828E ADDITN
690E   ADC #$0E
==8290 NEXT1
88     DEY
D0E5   BNE LOOPDV
B003   BCS LAST1
690E   ADC #$0E
18     CLC
==8298 LAST1
2E1045 ROL ANSWER

;DISPLAY DATA

AC1045 LDY ANSWER
88     DEY
B1FE   LDA (LOC),Y
2005EF JSR OUTD
AD1745 LDA PLIER
8D1245 STA MPLIER
==82AA
C90E   CMP #$0E
F009   BEQ DONE
EE1745 INC PLIER
EE1245 INC MPLIER
4C5B82 JMP MULTIP

==82B7 DONE
60     RTS

;DELAY/ERROR SUB

==82B8 DELERP
20C582 JSR DELQUE
200A83 JSR ERROR
20C582 JSR DELQUE
200A83 JSR ERROR
60     RTS

;START?/DELAY 1/2 S

==82C5 DELQUE
2070D1 JSR START
209883 JSR DDD
2070D1 JSR START
209883 JSR DDD

2070D1 JSR START
209883 JSR DDD
==82D7
2070D1 JSR START
209883 JSR DDD

2070D1 JSR START
209883 JSR DDD
2070D1 JSR START
209883 JSR DDD

==82E9
2070D1 JSR START
209883 JSR DDD
2070D1 JSR START
209883 JSR DDD

2070D1 JSR START

;REDUCTION REG. SET?

AD0D45 LDA DRF
==82FB
F005   BEQ BUTTON
==82FD AWAY
68     PLA
68     PLA
68     PLA
68     PLA

;GO REDUCE DATA

60     RTS

;FUNCTION BUTTON ?

==8302 BUTTON
AD0193 LDA FUNC
C9FE   CMP #$FE
F0F4   BEQ AWAY

60     RTS

;RESOLVER ANGL ERR ?

==830A ERROR
AD0448 LDA RAE
F02B   BEQ CASS

;DISPLAY RAE

A000   LDY #$00

==8311 LOADAE
B91F83 LDA EMSG1,Y
C93B   CMP #';'
F010   BEQ SUB1A
2005EF JSR $EF05 OUT
D
C8     INY
4C1183 JMP LOADAE

==831F EMSG1
5245   BYT 'RESOLVE
R ANGLE ERROR;'

==8334
;JUMP TO SUBROUTINE

==8334 SUB1A
20C582 JSR DELQUE

;CLEAR DISPLAY

20F0E9 JSR $E9F0 CRL
F

;CASSETTE TAPE ERR ?

==833A CASS
AD0748 LDA CTE
F02B   BEQ FULL
```

Figure A.8-3 Contour Display Program Listing continued

```
;DISPLAY CASERR

A000   LDY #$00

==8341 LOADCE
B94F83 LDA EMSG2,Y
C93B   CMP #';'
F01C   BEQ SUB1B
2005EF JSR $EF05 OUT
D
C8     INY
4C4183 JMP LOADCE

==834F EMSG2
4341   .BYT 'CASSETT
E TAPE ERROR ;'

==8364
;JUMP TO SUBROUTINE

==8364 SUB1B
200582 JSR DELQUE

;CLEAR DISPLAY

20F0E9 JSR $E9F0 CRL
F

;TAPE FULL CONDITION

==836A FULL
AD0A48 LDA TFR
F028   BEQ OUT

;DISPLAY FULERR

A000   LDY #$00

==8371 LOADTF
B97F83 LDA EMSG3,Y
C93B   CMP #';'
F01C   BEQ SUB1C
2005EF JSR $EF05 OUT
D
C8     INY
4C7183 JMP LOADTF

==837F EMSG3
5441   .BYT 'TAPE FU
LL-EXCHANGE ;'

==8394
;JUMP TO SUBROUTINE

==8394 SUB1C
200582 JSR DELQUE

==8397 OUT
60     RTS.

;DELAY; 1/2 SECOND

==8398 DDD
A2D9   LDX #$D9

==839A LXL
A0FF   LDY #$FF

==839C LYL
EA     NOP
EA     NOP
88     DEY
D0FB   BNE LYL
CA     DEX
D0F6   BNE LXL

60     RTS
```

Figure A.8-4 Contour Display Program Listing continued

A.9 Starting Point Program

The Starting Point Program enables the coal face measurement system to produce automatic repeatable operations. This program is accessed from the BASIC routine and the Contour Display routine. Whenever the mining machine returns to its' original position this program directs the system to normalize itself for the acquisition of new data.

Figure A.9-1 shows the flowchart while Figure A.9-2 presents the actual program listing.

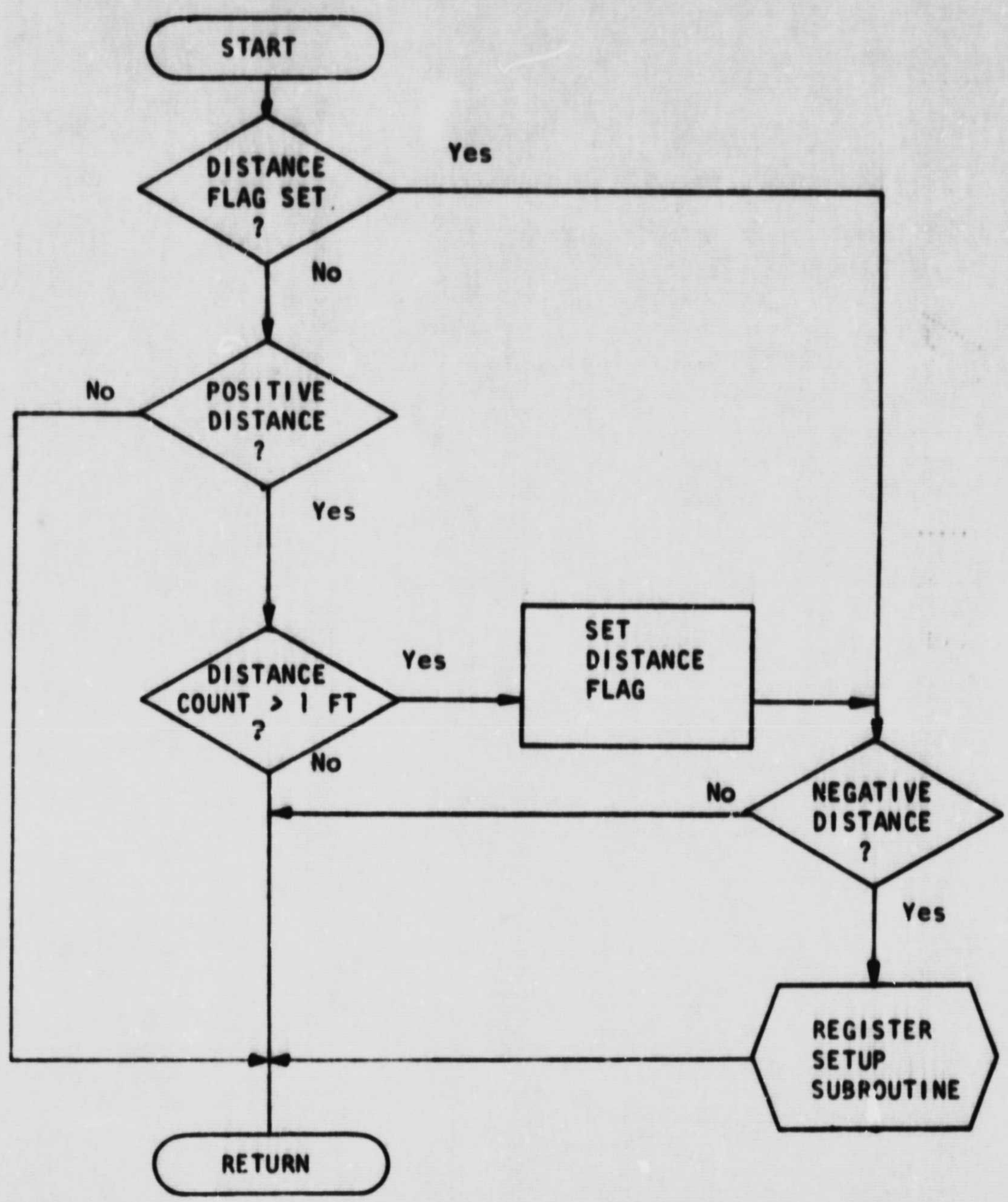


Figure A.9-1 Starting Point Program Flowchart

```
        *=$8170
---------------------
ZERO POSITION DETECT


==8170
;PROGRAM CONSTANTS

==8170 ZFR=$4514

==8170 SCODE=$480D

==8170 NORMAL=$DB50

==8170 PCRR=$9E0C

==8170 IFRR=$9E0D

==8170 IERR=$9E0E

==8170 DISTPA=$9F00

==8170 DISTPB=$9F02


;BACK TO START ?
;ZERO FLAG SET ?

AD1445 LDA ZFR
D013   BNE ZZZ

;POSITIVE DISTANCE ?

AD029F LDA DISTPB
C920   CMP #$20
B041   BCS NOPE

;GREATER THAN 1 FT ?

AD009F LDA DISTPA
C908   CMP #$08
==8181
903A   BCC NOPE

;YES; SET ZERO FLAG

A901   LDA #$01
8D1445 STA ZFR

;NEGATIVE DISTANCE ?

==8188 ZZZ
AD029F LDA DISTPB
C920   CMP #$20
902E   BCC NOPE

;DISABLE NMI

A902   LDA #$02
8D0E9E STA IERR

;SET BASIC STRT CODE

A901   LDA #$01
8D0D48 STA SCODE

==8199
;RESTORE SYSTEM

2050DB JSR NORMAL

;RESET DISTANCE

A9EC   LDA #$EC
8D0C9E STA PCRR
A9CC   LDA #$CC
8D0C9E STA PCRR

;ENABLE NMI

A982   LDA #$82
8D0D9E STA IFRR
==81AB
8D0E9E STA IERR

;SET DISTANCE

A9CE   LDA #$CE
8D0C9E STA PCRR
A9CC   LDA #$CC
8D0C9E STA PCRR

;RESET ZERO FLAG

A900   LDA #$00
8D1445 STA ZFR

==81BD NOPE
60     RTS
```

Figure A.9-2 Starting Point Program Listing

A.10 Utility Programs

The following are small programs which are used by the measurement system during normal operation:

1) Run Number Program

The Run Number Program enables the EPROM Programmer Program to fetch and save the run number register.

2) Arc-Tangent Program

The Arc-Tangent Program allows the BASIC routine to calculate the ARCSIN (x) function used in its' data reduction section.

3) EPROM Programmer Program

The EPROM Programmer Program enables the system to permanently retain the initial run data and record the latest run number. Thus a battery is not required to back-up the system to retain this information.

4) EPROM Reader Program

The EPROM Reader Program is a dedicated routine to fetch information from the system EPROM.

Listings for these four programs are presented as Figure A.10-1 through A.10-4.

```
        *=$8100
---------------------
EPROM SETUP FOR RMH

==8100 HCOUNT=$450E

==8100 LCOUNT=$450F

==8100 RMLHIB=$4516

;SETUP FOR DRAM USE

AD0E45 LDA HCOUNT
8502   STA $02

A902   LDA #$02
8500   STA $00
8503   STA $03

A903   LDA #$03
8504   STA $04

A948   LDA #$48
==8111
8501   STA $01
8505   STA $05

60     RTS
```

```
        *=$8150
---------------------
EPROM SETUP FOR RML

==8150
;SETUP FOR DRAM USE

AD0F45 LDA LCOUNT
8502   STA $02

AD1645 LDA RMLHIB
8503   STA $03

A901   LDA #$01
8500   STA $00

A902   LDA #$02
==8160
8504   STA $04

A948   LDA #$48
8501   STA $01
8505   STA $05

60     RTS
```

Figure A.10-1 Run Number Program Listing

```
<M>=2F80 0B 76 B3 83
< > 2F84 BD D3 79 1E
< > 2F88 F4 A6 F5 7B
< > 2F8C 83 FC B0 10
< > 2F90 7C 0C 1F 67
< > 2F94 CA 7C DE 53
< > 2F98 CB C1 7D 14
< > 2F9C 64 70 4C 7D
< > 2FA0 B7 EA 51 7A
< > 2FA4 7D 63 30 88
< > 2FA8 7E 7E 92 44
< > 2FAC 99 3A 7E 4C
< > 2FB0 CC 91 C7 7F
< > 2FB4 AA AA AA 13
< > 2FB8 81 00 00 00
</> 2FBC 00
```

```
/24
 2FBD A5 LDA AE
 2FBF 48 PHA
 2FC0 10 BPL 2FC5
 2FC2 20 JSR CCB8
 2FC5 A5 LDA A9
 2FC7 48 PHA
 2FC8 C9 CMP #81
 2FCA 9[illegible] [illegible] 2FD3
 2FCC [illegible] [illegible]A #FB
 2FCE [illegible]0 [illegible] #C6
 2FD0 [illegible] [illegible]SR C84E
 2FD3 A9 LDA #80
 2FD5 A0 LDY #2F
 2FD7 20 JSR CD44
 2FDA 68 PLA
 2FDB C9 CMP #81
 2FDD 90 BCC 2FE6
 2FDF A9 LDA #4E
 2FE1 A0 LDY #CE
 2FE3 20 JSR C58F
 2FE6 68 PLA
 2FE7 10 BPL 2FEC
 2FE9 4C JMP CCB8
 2FEC 60 RTS
```

Figure A.10-2 Arc-Tangent Program Code

```
<M>=D800 A0 20 A2 02
< > D804 A9 CC D0 06
< > D808 A0 02 A2 20
< > D80C A9 EC 84 11
< > D810 8F 10 85 12
< > D814 A0 03 B9 1B
< > D818 D9 99 06 00
< > D81C 88 10 F7 A9
< > D820 00 48 28 A2
< > D824 13 AD FD FF
< > D828 C9 8B F0 0A
< > D82C A2 0F C9 E0
< > D830 F0 07 A2 17
< > D834 D0 03 20 86
< > D838 8B 86 0E A9
< > D83C D9 85 0F A0
< > D840 03 B1 0E 99
< > D844 0A 00 88 10
< > D848 F8 A0 00 A9
< > D84C 1F 91 0C A9
< > D850 D9 C8 91 0C
< > D854 A0 0E A9 7F
< > D858 91 06 91 08
< > D85C A0 0D A9 FF
< > D860 91 06 91 08
< > D864 A0 0E A9 A0
< > D868 91 06 A2 00
< > D86C A0 0C A5 12
< > D870 91 06 A9 CC
< > D874 91 08 A9 FF
< > D878 A0 02 91 06
< > D87C A0 03 91 06
< > D880 91 08 A0 01
< > D884 A5 02 91 06
< > D888 A0 00 A5 03
< > D88C 91 06 A1 00
< > D890 A0 01 91 08
< > D894 A9 50 A0 08
< > D898 91 06 A9 C3
< > D89C A0 09 91 06
< > D8A0 A5 12 A4 03
< > D8A4 C0 08 30 02
< > D8A8 05 11 A0 0C
< > D8AC 91 06 45 10
< > D8B0 91 06 C0 0C
< > D8B4 F0 FC A0 0C
< > D8B8 A9 EC 91 08
< > D8BC A0 64 88 D0
< > D8C0 FD A0 0C A5
< > D8C4 10 C9 20 D0
< > D8C8 04 51 06 91
< > D8CC 06 A9 00 A0
< > D8D0 03 91 08 A0
< > D8D4 01 B1 08 C1
< > D8D8 00 F0 03 4C
< > D8DC 06 D9 E6 00
< > D8E0 D0 07 E6 01
< > D8E4 D0 03 4C 06
< > D8E8 D9 A5 05 C5
< > D8EC 01 D0 09 A5
< > D8F0 04 C5 00 D0
< > D8F4 03 4C 06 D9
< > D8F8 E6 02 D0 07
< > D8FC E6 03 D0 03
< > D900 4C 06 D9 4C
< > D904 6A D8 A0 0C
< > D908 A9 EE 91 08
< > D90C 60 00 00 6D
< > D910 E1 00 A4 35
< > D914 80 7E A6 05
< > D918 1C FE 17 00
< > D91C 70 00 78 B1
< > D920 06 45 10 91
< > D924 06 A0 0D B1
< > D928 06 91 06 40
```

Figure A.10-3 EPROM Programmer Program Code

```
<M>=DA00 A0 20 A2 02
< > DA04 A9 CC D0 06
< > DA08 00 00 00 00
< > DA0C 00 00 84 11
< > DA10 86 10 85 12
< > DA14 A0 03 B9 1B
< > DA18 DB 99 06 00
< > DA1C 88 10 F7 A9
< > DA20 00 48 28 A2
< > DA24 13 AD FD FF
< > DA28 C9 8B F0 0A
< > DA2C A2 0F C9 E0
< > DA30 F0 07 A2 17
< > DA34 D0 03 20 86
< > DA38 8B 86 0E A9
< > DA3C DB 85 0F A0
< > DA40 03 B1 0E 99
< > DA44 0A 00 88 10
< > DA48 F8 EA EA EA
< > DA4C EA EA EA EA
< > DA50 EA EA EA EA
< > DA54 A0 0E A9 7F
< > DA58 91 06 91 08
< > DA5C A0 0D A9 FF
< > DA60 91 06 91 08
< > DA64 EA EA EA EA
< > DA68 EA EA A2 00
< > DA6C A0 0C A5 12
< > DA70 91 06 EA EA
< > DA74 EA EA A9 FF
< > DA78 A0 02 91 06
< > DA7C A0 03 91 06
< > DA80 EA EA A0 01
< > DA84 A5 02 91 06
< > DA88 A0 00 A5 03
< > DA8C 91 06 4C CD
< > DA90 DA 00 00 00
< > DA94 00 00 00 00
< > DA98 00 00 00 00
< > DA9C 00 00 00 00
< > DAA0 00 00 00 00
< > DAA4 00 00 00 00
< > DAA8 00 00 00 00
< > DAAC 00 00 00 00
< > DAB0 00 00 00 00
< > DAB4 00 00 00 00
< > DAB8 00 03 00 00
< > DABC 00 00 00 00
< > DAC0 00 00 00 00
< > DAC4 00 00 00 00
< > DAC8 00 00 00 00
< > DACC 00 A9 00 A0
< > DAD0 03 91 08 A0
< > DAD4 01 B1 08 81
< > DAD8 00 EA EA EA
< > DADC EA EA E6 00
< > DAE0 D0 07 E6 01
< > DAE4 D0 03 4C 06
< > DAE8 DB A5 05 C5
< > DAEC 01 D0 09 A5
< > DAF0 04 C5 00 D0
< > DAF4 03 4C 06 DB
< > DAF8 E6 02 D0 07
< > D9FC E6 03 D0 03
< > DB00 4C 06 DB 4C
< > DB04 6A DA A0 0C
< > DB08 A9 EE 91 08
< > DB0C 60 00 00 6D
< > DB10 E1 00 A4 00
< > DB14 00 00 00 00
< > DB18 00 00 00 00
< > DB1C 70 00 78 00
</> DB20 00
```

Figure A.10-4 EPROM Reader Program Code

A.11 System Memory Map

The memory usage for the coal face measurement system is presented as Figure A.11-1. It shows the allocation for the 64,000 possible memory addresses. The system programs are found in two groups. The following routines begin at hexidecimal address 5000:

1) BASIC Program
2) Input/Store Program
3) Data Transfer Program

The remaining routines begin at hexidecimal address DOOO:

1) Input/Output Setup Program
2) Run Number Program
3) Starting Point Program
4) Display Program
5) Arc-Tangent Program
6) EPROM Programmer Program
7) EPROM Reader Program
8) Register Setup Program

8000 unused address locations provide room for future program expansion.

SIZE (BYTES)	ADDRESS (HEX)	DESCRIPTION
768	0000-02FF	AIM 6500 RAM
7424	0300-1FFF	BASIC Routine Variables
4K	2000-2FFF	Available for use
4K	3000-3FFF	Temporary Data Strorage
4K	4000-4FFF	System Registers
6144	5000-67FF	BASIC PROGRAM
1K	6800-6BFF	INPUT/STORE PROGRAM
1K	6C00-6FFF	DATA TRANSFER PROGRAM
4K	7000-7FFF	EPROM Programmer
4K	8000-8FFF	Available for use
4K	9000-9FFF	Input/Output Ports
4K	A000-AFFF	AIM 6500 Peripherals
8K	B000-CFFF	AIM 6500 BASIC
256	D000-D0FF	INPUT/OUTPUT SETUP PROGRAM
112	D100-D16F	RUN NUMBER PROGRAMS
144	D170-D1FF	STARTING POINT PROGRAM
1K	D200-D5FF	DISPLAY PROGRAM
512	D600-D7FF	ARC-TANGENT PROGRAM
512	D800-D9FF	EPROM PROGRAMMER PROGRAM
336	DA00-DB4F	EPROM READER PROGRAM
1200	DB50-DFFF	REGISTER SETUP PROGRAM
8K	E000-FFFF	AIM 6500 Monitor
64K		

Figure A.11-1 System Memory Map