To the Graduate Council:

I am submitting herewith a thesis written by Leland Wayne Barker entitled "Development of a Microcomputer Based Flowmeter for Liquid Volume Flow in Partially Full Pipes." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Donald W. Bouldin

Donald W. Bouldin, Major Professor

We have read this thesis
and recommend its acceptance:

Robert W. Rochelle

Robert E. Bodenheimer

Accepted for the Council:

L. Evans Roth

Vice Chancellor
Graduate Studies and Research

DEVELOPMENT OF A MICROCOMPUTER BASED FLOWMETER FOR LIQUID VOLUME FLOW IN PARTIALLY FULL PIPES

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Leland Wayne Barker

June 1980

ACKNOWLEDGEMENTS

The author wishes to thank the members of his committee, Drs. D. W. Bouldin, R. E. Bodenheimer and R. W. Rochelle. Special thanks go to Dr. Bouldin who was constantly available to advise on the writing of the thesis.

Thanks are also due to two people with whom the author has worked. Mr. Bill Osborne and Mr. Darrell Corpening were greatly appreciated for their assistance in learning to use both the Intel MDS 800 development system and the assembly language subprograms available. Their assistance greatly facilitated the writing of the computer program.

Final thanks must go to the author's wife, Chris, for her encouragement and helpful comments.

ABSTRACT

A flowmeter was designed, developed and constructed to measure total volume flow in partially full pipes because a reliable, accurate flowmeter was not commercially available. The flowmeter system uses a pitot tube type sensor with a differential pressure transmitter and a bubbler tube with a differential pressure transmitter to obtain flow velocity and liquid level information. A microcomputer performs all the flow calculations from this information. The completed system exceeds all desired performance specifications for reliability, accuracy, response time and flexibility.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background Information

Total volume flow measurement has always been a difficult task. Historically, this measurement has been almost impossible to make in pipelines running partially full. A measurement of both flow velocity and liquid level must be made to calculate total volume flow for this condition. No acceptable method has existed for making the flow velocity measurement under this condition. Nevertheless, volume flow measurement can be very important to process control and is sometimes necessary to satisfy Environmental Protection Agency regulations.

One instrument manufacturer has recently developed a system to make this measurement. The author's company bought three of these units for trial evaluation, partially because the instrument looked promising, and partially because no other instruments were commercially available. The instrument uses a magnetic sensor to measure flow velocity and uses a bubbler system incorporating a pressure transmitter to measure the liquid level. The total volume flow calculations are made by hardwired logic using discrete components and integrated circuits.

These three flowmeters were installed in different plant locations. However, all flowmeters experienced hardware and design problems from the start. Engineering and maintenance personnel were never able to make the system perform satisfactorily, even with the assistance of manufacturer service representatives. When this became evident, supervision decided to develop a system in house.

Analysis of the Commercial Flowmeter

Several problem areas had become evident in the purchased flowmeters. A major problem was the failure of discrete components and integrated circuits in the hardwired electronic logic that performs the flow calculations. This problem recurred often. When failures occurred, these units were either removed and shipped back to the manufacturer for rapair or factory personnel were sent for on site repair, since the instruments were still under warranty. Down time for the instruments due to this problem approached 50% the first year. As a result, these three systems had not operated simultaneously during the entire first year.

Another class of problems resulted from the inaccuracy of the sensor used to measure flow velocity. In this design the single point sensor develops a magnetic field only in the immediate area of the probe. The flow velocity through this field causes a voltage to be developed across two points in the probe. So, in effect, this sensor measures flow velocity in only one small area of the pipe in contrast to the Annubar method described in Chapter II. The measured velocity may or may not be close to the average fluid velocity in the pipe since there are different flow velocities in different areas of a pipe. If the measured velocity is not approximately the average fluid velocity, there will be a large error in the total flow output. This results because the total volume flow is a function of fluid velocity. Accurate placement of this probe in order to obtain the average velocity would require a detailed knowledge of the flow profile of the pipe. In addition, any type of obstruction in the pipe alters this flow profile and decreases the accuracy of the flow measurement. Coatings on the probe may also decrease system accuracy by altering the magnetic field in the immediate vicinity of the probe.

The manufacturer's method of measuring liquid level by using a bubbler tube with a differential pressure transmitter is satisfactory. Total accuracy of a bubbler system is within ± 0.5% of full scale range since the accuracy of most differential pressure transmitters, the measuring instrument, is within 0.25% of full scale range. The bubbler tube adds very little inaccuracy to the system since the pressure drop is small when air velocity is small.

As a final consideration, this system is not very flexible. The flowmeter utilizes custom built logic, so the electronics are designed for the particular location in which they are to be used. Transferring the system to other locations with greatly different flow velocities and pipe sizes may require component changes as well as electronic adjustments since the pipe radius and the flow velocity are used in the flow calculations. The new pipe radius must be substituted into the logic that performs the flow calculations. Significantly different flow velocities will cause greatly different voltage levels on the flow sensor. Scaling the altered voltage level on the sensor will require a change in the flow constant designed into the electronics.

Chapter Content

The general equation derivation, the specific equation derivation and the sensors used are covered in Chapter II. The sensors are included in this chapter because their outputs must be included in the derivation of the specific equation. The hardware system considerations are explained in Chapter III. Desirable qualities are arrived at for a new flowmeter and the choice of hardware for the new system is discussed in this chapter. Finally, the actual hardware layout is given and block diagrams are included. An explanation is given in Chapter IV for the particular choice of software language. The general program flowchart is also included and explained. Chapter V details the performance of the new system and compares it to the old system and to desired performance levels. Chapter VI presents a summary and conclusions of the project.

CHAPTER II

DESCRIPTION OF THE PROCESS

A theoretical analysis of the commercial method for obtaining flow is given for the particular application. Descriptions of the sensors used to obtain liquid level information and velocity information, both of which are used in application of the theoretical treatment, are also included.

General Flow Equation

The volume flow rate in a partially filled pipe is the volume of liquid that flows past any point in unit time and is written as $\frac{d(vol)}{dt}$. This can be written as $\frac{d(l \cdot area)}{dt}$ since the volume of any figure with a constant cross sectional area is equal to the length (l) times the cross sectional area. The average distance that a unit moves in unit time is l. For a constant cross sectional area, the volume rate $\frac{d(vol)}{dt}$ can be rewritten as area x $\frac{dl}{dt}$ where $\frac{dl}{dt}$ equals the average velocity of the fluid. One can then write

$$\text{flow} = \frac{d(vol)}{dt} = \text{velocity x area.} \qquad (1)$$

Cross Sectional Area Derivation

In a cylindrical pipe, the cross sectional area can be calculated from the radius of the pipe and the depth of the liquid. A formula for the cross sectional area is developed for Case 1 where the liquid depth is less than R, the pipe radius, and for Case 2, where the depth is greater than R.

From Figure 1, it is obvious for Case 1, $D < R$, that the total cross-sectional area of the fluid is equal to the cross-sectional area of the bottom half of the pipe minus the sum of the area of the triangle and the areas of the two sectors included in the lower semicircular area. This is written as

$$A_1 = \text{bottom semicircle} - (\text{triangle} + 2 \text{ sectors})$$

Substituting in the area formulas gives

$$A_1 = \frac{\pi R^2}{2} - [(R - D)\sqrt{R^2 - (R - D)^2} + 2(\frac{\theta}{2\pi})(\pi R^2)]$$

$$= \frac{\pi R^2}{2} - (R - D)\sqrt{R^2 - (R^2 - 2RD + D^2)} - \theta R^2$$

Since $\sin\theta = \frac{R - D}{R}$, $\theta = \sin^{-1}(\frac{R - D}{R})$

$$A_1 = \frac{\pi R^2}{2} - (R - D)\sqrt{2RD - D^2} - R^2 \sin^{-1}(\frac{R - D}{R})$$

$$= \frac{\pi R^2}{2} - [(R - D)\sqrt{D(2R - D)} + R^2 \sin^{-1}(\frac{R - D}{R})], \quad (0 \leq \frac{R - D}{R} \leq 1) \tag{2}$$

From Figure 2, it is obvious for Case 2, $D > R$, that the total cross-sectional area of the fluid is equal to the cross-sectional area of the bottom half of the pipe plus the sum of the area of the triangle and the areas of the sectors included in the upper semicircular area. This is written as

$$A_2 = \text{bottom semicircle} + (\text{triangle} + 2 \text{ sectors})$$

Substituting in the area formulas gives

$$A_2 = \frac{\pi R^2}{2} + [(D - R)\sqrt{R^2 - (D - R)^2} + 2(\frac{\theta}{2\pi})(\pi R^2)]$$

$$= \frac{\pi R^2}{2} + (D - R)\sqrt{R^2 - (D^2 - 2RD + R^2)} + \theta R^2$$

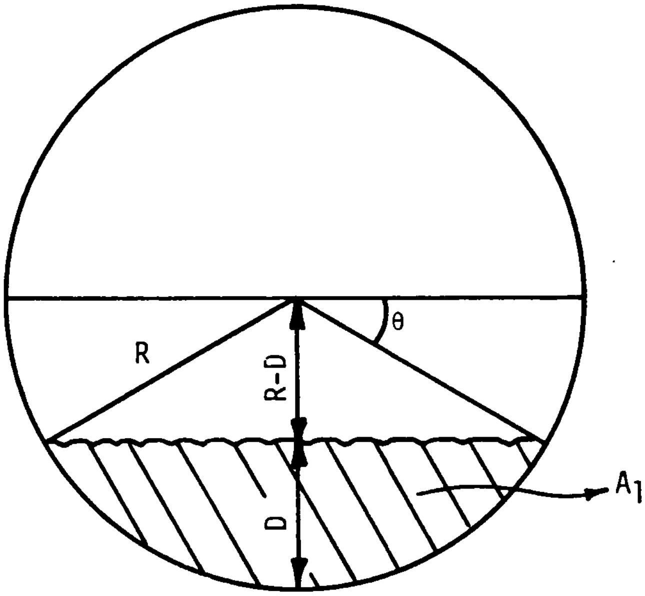


Figure 1. Pipe Less than Half Full (D < R)

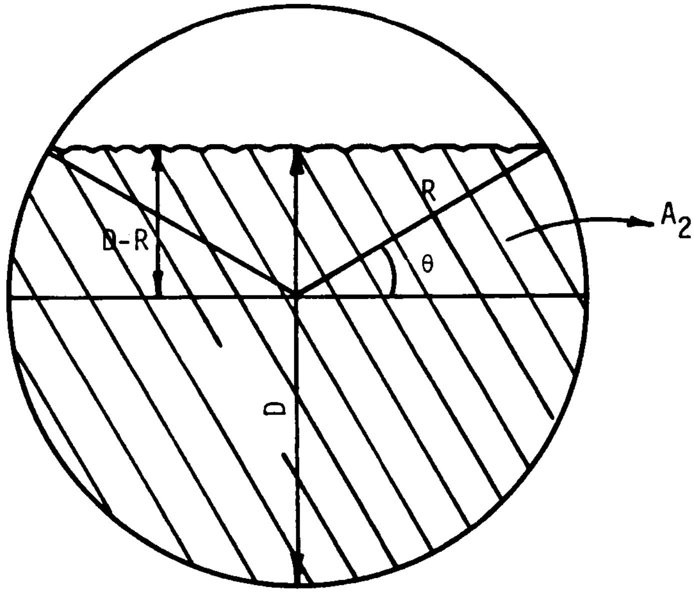


Figure 2. Pipe More than Half Full (D > R)

Again, $\theta = \sin^{-1} (\frac{R - D}{R})$ so

$$A_2 = \frac{\pi R^2}{2} + (D - R) \sqrt{2RD - D^2} + R^2 \sin^{-1} (\frac{D - R}{R})$$

Since sin (-x) = - sin x and D - R = - (R - D)

$$A_2 = \frac{\pi R^2}{2} - (R - D) \sqrt{D(2R - D)} - R^2 \sin^{-1} (\frac{R - D}{R})$$

$$= \frac{\pi R^2}{2} - [(R - D) \sqrt{D(2R - D)} + R^2 \sin^{-1} (\frac{R - D}{R})], (-1 \leq \frac{R - D}{R} \leq 0)$$

(3)

Equation 3 from Case 2 is identical to Equation 2 from Case 1 so the area can be written as

$$A = \frac{\pi R^2}{2} - [(R - D) \sqrt{D(2R - D)} + R^2 \sin^{-1} (\frac{R - D}{R})], (-1 \leq \frac{R - D}{R} \leq 1)$$

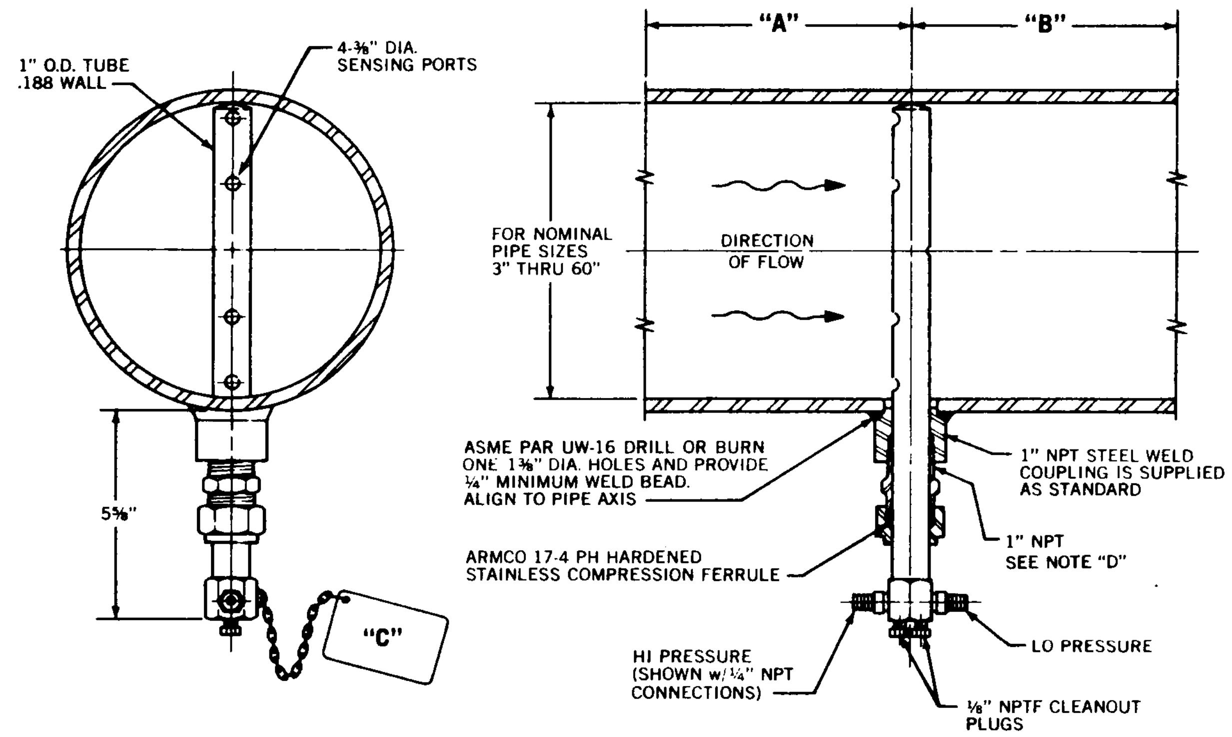
(4)

The equations for the area of a semicircle, a triangle, and a sector of a circle are included in many mathematical tables such as CRC Standard Mathematical Tables. [1] The most important point in the derivations was recognizing that both total area equations could be written in identical form. This was an original contribution that greatly simplified flow computations.

Sensors

The liquid level, D, may be measured using a bubbler tube type differential pressure system. Liquid level is measured extensively by this method in the chemical industries. It has proven extremely reliable and accurate. Other acceptable methods of determining liquid levels exist but standard instrumentation is not available to convert the measurement to an electrical signal. In this method the low pressure connection of the transmitter is left open to the atmosphere. A length of tubing is then connected by one end to the high pressure port of a differential pressure transmitter. The other end of the tubing is positioned near the bottom of the pipe in which the level measurement is to be made and left open. A connection is made in this tubing so that a small flow of air may be introduced. Due to this air flow, the pressure in the tubing will increase until it exactly equals the external pressure on the open end of the tubing. At this pressure, any additional air flow into the tubing will bubble out the end. The pressure in the tubing is directly proportional to the height of the column of water above the open end of the tubing. Therefore, the pressure detected by the differential pressure transmitter is directly proportional to the liquid level in the pipe.

The fluid velocity may be measured by an Annubar primary sensor manufactured by Dieterich Standard Corporation (see Figure 3). [2] Using the Annubar sensor eliminates sensor problems encountered with the commercial flowmeters. The advantages of using this sensor are given in Chapter III, Hardware System Considerations. This sensor can be described as two pitot tubes that use the concept of annular averaging. [3] The input probe is designed to measure the average velocity of the flowing fluid. Four ports are located in an upstream direction, facing the direction of the flow. The pressures at these four ports are averaged by an interpolating tube or pitot tube inside the sensor. This pitot tube, which points toward the ports, measures the pressure inside the sensor caused by flow velocity at the four ports. The



"A"—6 OR MORE PIPE DIAMETERS IS RECOMMENDED FOR UPSTREAM SIDE AFTER VALVES, ELBOWS, ETC. SEE FORM E-79 BEFORE INSTALLATION.

"B"—3 TO 4 PIPE DIAMETERS IS RECOMMENDED FOR DOWNSTREAM SIDE.

"C"—PERMANENT RUSTPROOF METAL TAG SHOWING MIN., NORM. & MAX. DESIGNED FLOWS, METER READINGS FOR DESIGNED FLOWS, TAG NO., LINE SIZE, SERIAL NO., & METERED FLUID.

"D"—FOR PIPES OF NON-WELDABLE MATERIAL, TAP THE PIPE WALL AND THREAD DIRECTLY INTO THE WALL.

"E" - THE ANNUBAR IS PLACED HORIZONTALLY IN THE PIPE BELOW THE LEVEL OF FLOW. NO PROVISION IS MADE IN THIS SYSTEM FOR AN ALARM WHEN THE SENSOR IS UNCOVERED.

Figure 3. Annubar Flow Sensor

Source: 751 through 756 Annubar Flow Elements Form No. E-167. "Annubar Primary Flow Sensors for Gas, Liquid or Steam," Pub. E-100, 1974. Dieterich Standard Corp., Boulder, Colo.

difference in pressure between that in the interpolating tube and a downstream pointing pitot tube is the average pressure difference of the total flow.

This pressure difference is measured by using another differential pressure transmitter. Tubing is connected between the interpolating tube and the high pressure port of the transmitter. Another length of tubing is connected between the downstream element of the Annubar and the low pressure port of the transmitter. Air is bubbled out the interpolating tube and the downstream element in a manner similar to that used to measure liquid level. The transmitter detects difference in tubing air pressures, which is exactly equal to the pressure difference in the sensing element.

Total flow calculations can then be made using these differential pressure measurements which yield depth and velocity information.

Specific Flow Equation

For two pitot tubes, one facing upstream and one downstream, $V = K\sqrt{dp}$. [4] dp is the differential pressure between the two tubes, V is the flow velocity of the liquid and K is a constant for the particular sensor. For our 8-inch Annubar sensor

$$Q_n = SND^2 F_a F_m F_p \sqrt{\frac{G_F}{G_l}} \sqrt{dp} = \frac{d(vol)}{dt} \quad [2]$$

Q_n = volume flow rate in cfm for a full pipe

S = constant for an Annubar for 8" pipe = 0.72734 = $K_g F_\nu$

N = grouped constants depending on units for Q_n = 0.7576

D = diameter of pipe in inches

F_a = expansion-contraction factor for pipe $\approx$ 1

F_m = manometer correction factor $\approx$ 1

F_p = correction for compressibility of flowing liquid = 1 for water

G_p = specific gravity of liquid at flowing temp. as compared to water at 60°F $\approx$ 1 in our case

G_1 = specific gravity of liquid at 60°F as compared to water at 60°F = 1

We can derive velocity by noting $V = \frac{d(vol)}{dt \times area}$.

Then

$$V(ft/sec) = \frac{Q_h(cfm)}{60[\pi(\frac{1}{3})^2]} \approx \frac{SND^2}{60[\pi(\frac{1}{3})^2]} \sqrt{dp}$$

$$= \frac{(0.72734)(0.7576)8^2}{(60)\ [\pi(\frac{1}{3})^2]} \sqrt{dp} = 1.68 \sqrt{dp} \text{ ft/sec}$$

So from $V_{avg} = K \sqrt{dp}$ ft/sec,

we see $K \approx 1.68$ ft/(sec $\sqrt{\text{in } H_2O}$) for an Annubar in a full pipe.

Since the pipe cross-sectional area has been removed from the above equation, only S might be different for a partially full pipe where

$$S = K_g F_\nu \quad [3]$$

K_g is a geometrical correction factor for the Annubar and does not vary if the pipe is only partially full. F_ν is the velocity distribution factor in the pipe. For a full pipe, this factor is given by

$$F_\nu = \frac{2n^2}{(2n + 1)(n + 1)}$$

for a $\frac{1}{nth}$ power law turbulent flow.

For Annubar calculations, the pipe is assumed to follow a $\frac{1}{7th}$ power law or

$$\frac{V_i}{V_{max}} = \left(\frac{y_i}{R}\right)^{\frac{1}{7}}$$

V_i = velocity at the ith port

V_{max} = centerline velocity

y_i = distance from wall to ith hole

R = pipe radius

Any changes in K caused by deviations from this velocity distribution in a partially filled pipe will be corrected for by field calibrations of the dp transmitter since the changes can be lumped as some constant multiple of the differential pressure. When K is not equal to 1.68, $K = 1.68\ K_1$ so that $V_{avg} = 1.68\ K_1\ \sqrt{dp_1}$. Then

$$V_{avg} = 1.68\ \sqrt{K_1{}^2 dp_1}$$

$$= 1.68\ \sqrt{dp}\ \text{ft/sec.} \qquad (5)$$

From Equation 1, flow = velocity$_{(avg.)}$ x area and substituting Equation 4 for area and Equation 5 for velocity we get

$$\text{Flow} = 1.68\ \sqrt{dp}\ \left\{ \frac{\pi R^2}{2} - \left[(R - D)\ \sqrt{D(2R - D)} + R^2 \sin^{-1}\left(\frac{R - D}{R}\right)\right]\right\} \text{ft}^3/\text{sec}$$

when R and D are expressed in feet. We approximate arcsine by a series truncated after four terms

$$\sin^{-1}(x) \approx x + \frac{x^3}{6} + \frac{3}{40}\ x^5 + \frac{15}{336}\ x^7 \quad (-1 \le x \le 1)\ [1]$$

$$= x + 0.166\ x^3 + 0.075\ x^5 + 0.045\ x^7$$

so

$$\text{Flow} \approx 1.68 \sqrt{dp} \{ \frac{\pi R^2}{2} - [(R - D) \sqrt{D(2R - D)} + R^2 (\frac{R - D}{R}$$
$$+ 0.166 (\frac{R - D}{R})^3 + 0.075 (\frac{R - D}{R})^5 + 0.045 (\frac{R - D}{R})^7)]\} ft^3/sec \tag{6}$$

This calculation for total volume flow can be made using only the differential pressure information obtained by the Annubar and the depth information obtained by the bubbler system.

<u>Maximum Error Due to Truncated Series</u>

To find the maximum error in the truncated series we find the maximum of

$$F(x) = \sin^{-1} x - (x + \frac{x^3}{6} + \frac{3}{40} x^5 + \frac{15}{336} X^7)$$

$$\frac{d}{dx} [\sin^{-1} x - (x + \frac{x^3}{6} + \frac{3}{40} x^5 + \frac{15}{336} x^7) = 0$$

$$\frac{1}{\sqrt{1 - x^2}} - (1 + \frac{1}{2} x^2 + \frac{3}{8} X^4 + \frac{105}{336} X^6) = 0$$

$$1 - \sqrt{1 - x^2} (1 + \frac{1}{2} x^2 + \frac{3}{8} x^4 + \frac{105}{336} x^6) = 0$$

$$1 = \sqrt{1 - x^2} (1 + \frac{1}{2} x^2 + \frac{3}{8} x^4 + \frac{105}{336} x^6)$$

We square both sides

$$1 = (1 - x^2)(1 + x^2 + x^4 + x^6 + 0.453125 x^8 + 0.234375 x^{10} + 0.09765625 x^{12})$$
$$= 1 + x^2 + x^4 + x^6 + 0.453125 x^8 + 0.234375 x^{10} + 0.09765625 x^{12} - x^2$$
$$- x^4 - x^6 - x^8 -0.453125 x^{10} - 0.234375 x^{12} - 0.09765625 x^{14}$$
$$0 = - 0.546875 x^8 - 0.21875 x^{10} - 0.13671875 x^{12} - 0.09765625 x^{14}$$
$$= - x^8 (0.546875 + 0.21875 x^2 + 0.13671875 x^4 + 0.0976525 x^6)$$

There is no solution other than x = 0 which is a minimum since there is 0 error at this point.

$$f(0) = \sin^{-1}(0) - 0 = 0$$

f(x) must then be a maximum at the boundary points -1 and +1. In our equations $x = \frac{R - D}{R}$ $(-1 \leq x \leq 1)$

$$f(-1) = \sin^{-1}(-1) - [-1 + \frac{(-1)^3}{6} + \frac{3}{40}(-1)^5 + \frac{15}{336}(-1)^7]$$

$$f(-1) = -0.284487$$

$$\sin^{-1}(-1) = -1.570796$$

$$\% \text{ error} = \frac{-0.284487}{-1.570796} \times 100 = 18\% \text{ at } x = -1$$

$$f(1) = \sin^{-1}(1) - [1 + \frac{1}{6} + \frac{3}{40} + \frac{15}{336}]$$

$$f(1) = 0.284487$$

$$\sin^{-1}(1) = 1.570796$$

$$\% \text{ error} = \frac{0.284487}{1.570796} \times 100 = 18\% \text{ at } x = 1$$

However, the % error in area is much smaller. From

$$A = \frac{\pi R^2}{2} - [(R - D)\sqrt{D(2R - D)} + R^2 \sin^{-1}(\frac{R - D}{R})]$$

$$A \approx \frac{\pi R^2}{2} - \{(R - D)\sqrt{D(2R - D} + R^2 [\frac{R - D}{R} + 0.166 (\frac{R - D}{R})^3$$

$$+ 0.075 (\frac{R - D}{R})^5 + 0.045 (\frac{R - D}{R})^7]\}$$

For a full pipe where D = 2R, $x = \frac{R - D}{R} = -1$ the exact area

$$= \pi R^2 = 3.141593\ R^2.$$

In our approximation,

$$\text{area} \approx \frac{\pi R^2}{2} - [(-R)\ \sqrt{0} + R^2\ [-1 + 0.166(-1)^3 + 0.075(-1)^5 + 0.045(-1)^7]$$

$$= \frac{\pi R^2}{2} - R^2\ [-\ 1.286] = 2.856796\ R^2$$

$$\%\ \text{of full scale error} = \frac{(3.141593\ -\ 2.856796)\ R^2}{3.141593\ R^2}\ X\ 100 = 9.06\%$$

For an empty pipe where D = 0, $x = \frac{R - D}{R} = 1$ the exact area = 0. In our approximation,

$$\text{area} \approx \frac{\pi R^2}{2} - [R\ \sqrt{0}\ + R^2\ (1 + 0.166 + 0.075 + 0.045)]$$

$$= \frac{\pi R^2}{2} - R^2\ [1.286000]$$

$$= R^2\ [0.284796]$$

$$\%\ \text{of full scale error} = \frac{0.284796}{3.141593}\ \frac{R^2}{R^2} = 9.06\%$$

We see using only four terms restricts the maximum error to 9% in the calculations of the cross sectional area.

CHAPTER III

HARDWARE SYSTEM CONSIDERATIONS

System Requirements

The new system should meet several basic requirements. Overall system accuracy should be within 15% or less of full scale. This 15% accuracy represents a compromise between the accuracy desired and what we thought could be reasonably achieved. Time between consecutive total flow measurement outputs must be one minute or less. System downtime should be held to less than 10%. Coatings on the probe should have little effect on the total measurement. Finally, one system design should be suitable for all applications thus eliminating basic hardware modifications when applying the system.

Hardware Considerations

The first concern was with the type of flow sensor needed. This sensor should measure an area weighted average velocity in the pipe to obtain a highly accurate reading. By sensing at multiple points in the pipe, a nearly average flow velocity could be obtained.

Multiple magnetic sensors could be used with the flow averaging performed by electronics. However, this would require placement of multiple probes. The electronic calculation would also become more complex by including this averaging function. Finally, the inaccuracies caused by coatings and obstructions in the flow would not be eliminated but could be experienced by any of the sensors.

Use of the Annubar sensor eliminates all sensor problems previously mentioned. The Annubar has multiple ports so that an average flow velocity may be obtained. The averaging is done by the probe instead of by the flowmeter electronics thus simplifying the calculations. Since all the ports are located in one element, only a single probe must be placed and anchored in the stream. In addition, coatings will not affect the sensor unless one or more of the ports are completely blocked. Although an obstruction in front of any one port can cause an error in the velocity measurement, the total measurement will be only marginally affected since the pressure at any one port is averaged with the pressure at the three other ports.

A 20% pressure error caused by a flow disturbance at one port would be reduced to an overall error of 5%. This effect can be demonstrated very simply. In the case where the pressure is equal to 20" H_2O at all ports, the average is also 20" H_2O. If however a 20% error exists at one port, the pressure at that port will be 16" H_2O or 24" H_2O. The error at that port is 4" H_2O. When 16" H_2O is averaged with the pressure at the other three ports, the result is $\frac{(16 + 20 + 20 + 20)}{4}$ or 19" H_2O. When 24" H_2O is averaged with the pressure at the other three ports, the result is $\frac{(24 + 20 + 20 + 20)}{4}$ or 21" H_2O. In both cases the error is now 1" H_2O, 1/4 the original error, and is only a 5% error.

A differential pressure of approximately 100" of water is developed across the Annubar upstream and downstream connections at the maximum expected flow velocity of 16 ft/sec. [2] This is converted to a 4-20 milliamp direct current signal and transmitted by a differential pressure transmitter and bubbler tube arrangement as shown in Figure 4. This particular transmitter is a very dependable instrument that has been used successfully. Accuracy of this transmitter is 0.2% of full scale which is 0.2" water in our system. Liquid level in the pipe is also measured by using a bubbler tube and a differential pressure transmitter and is also shown in Figure 4. Transmitter accuracy is the same. Overall system accuracy depends on the accuracy of the transmitter calibrations.

Next, the author considered the electronic specifications required in order to meet total system specifications. First, the electronics must be capable of making the necessary flow calculations. One calculation and total volume flow output per minute is required. The electronics should be reliable, keeping downtime to a minimum. Since there are several applications for this flow meter, the instrument should be designed so that a similar unit can be used at any of the locations. System cost is also very important.

A digital system consisting of a National Semiconductor Rack Mounted Computer RMC 80/10, a Datel Systems model 800 analog to digital (A/D) converter and a Datel Systems Model 800 digital to analog (D/A) converter meets all these requirements. This is a representative digital system and descriptions of this system are included in the text for completeness. There are many other systems that could be used equally. The flow calculation can be easily programmed into this type unit. The basic unit could be used at any location with only minor programming changes required for each application. The maximum error in the calculation of the equations in Chapter II is 9% or less. This result was shown in Chapter II. Additional calculations that have been made but not included show that this error drops to 5% or less when the pipe is more than 5% or less than 95% full. The pipe liquid level should be between these limits 99% of the time, therefore overall accuracy will be much better than 9%. Accuracy of the analog to digital conversion by a Datel Systems Model 800 is within ± 0.025% of the full scale range. The

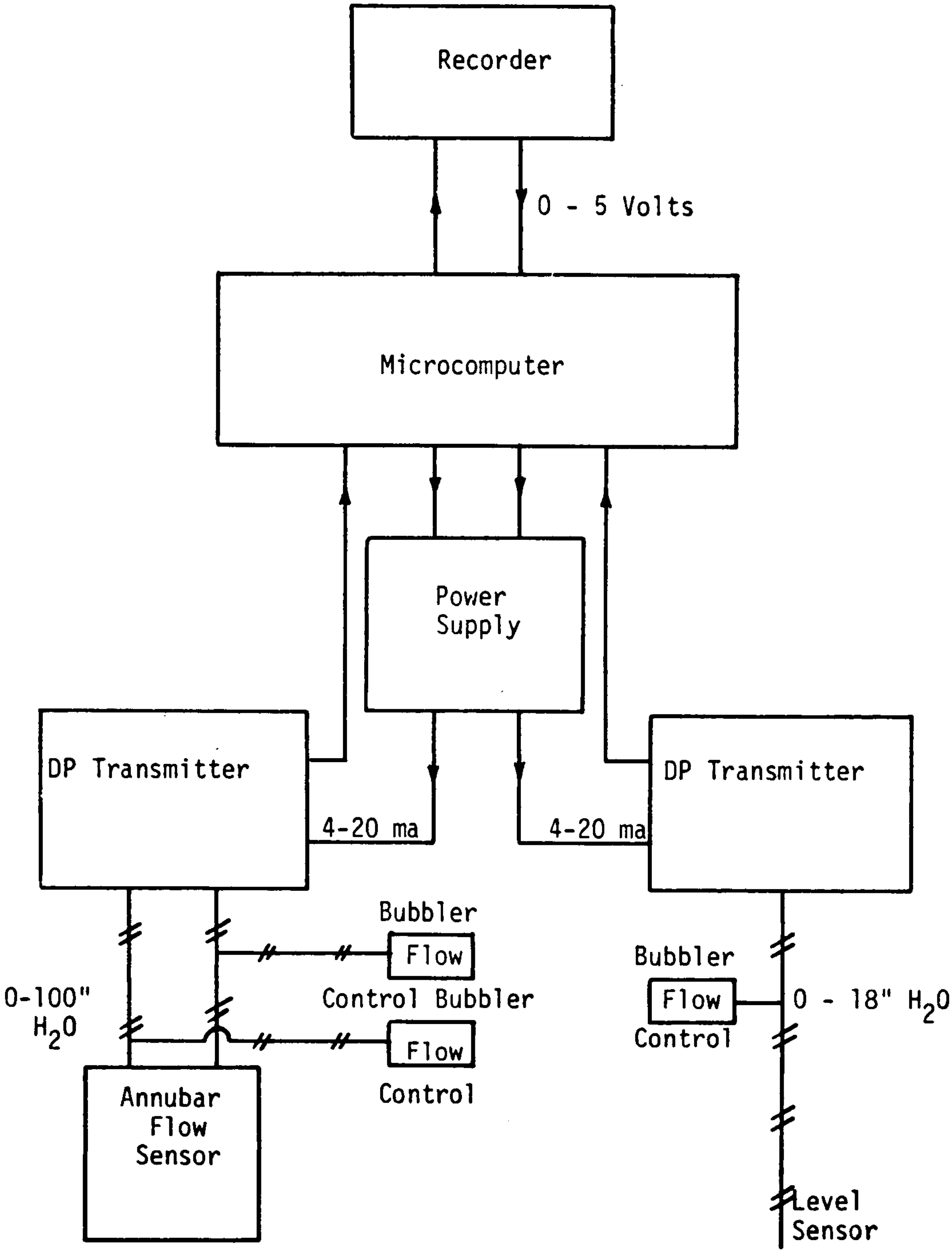


Figure 4. System Block Diagram

resolution is one part in 4096. Accuracy of the digital to analog conversion is $\pm \frac{1}{2}$ least significant bit over the entire output range when zero and full scale are adjusted correctly. This calculates to an accuracy within 0.011% of full scale range. Finally, the cost of the microcomputer system should be less than the cost of designing and building a hardwired system. The block diagram for the total system is shown in Figure 4. The system interconnection diagram is shown in Figure 5.

Differential Pressure Transmitter. The differential pressure unit employed uses a capacitance sensor. [5] The high and low pressures from the process connections are transmitted through isolating diaphragms and sealed fill fluid to a sending diaphragm in the center of the sensing element. This sensing diaphragm deflects in response to the differential pressure across it. This deflection is detected by fixed capacitor plates on both sides of the sensing diaphragm. The diaphragm is actually the second plate for both of these capacitors. Current output is proportional to this deflection.

RMC 80/10 Microcomputers. The National Semiconductor Rack Mounted Computer, RMC 80/10, is a packaged microcomputer system built around the BLC-80/10 Board Level Computer and is comprised of five functional units. [6] These units are a BLC-80/10 Board Level Computer, a BLC-910 System Monitor for loading, executing and debugging programs, a BLC-635 power supply, a BLC-604 four slot card cage, and front panel controls.

The BLC-80/10 is a self contained board level computer based on the INS8080A Microprocessor. [7] It is a complete computer including a Central Processor Unit (CPU), a serial communications interface,

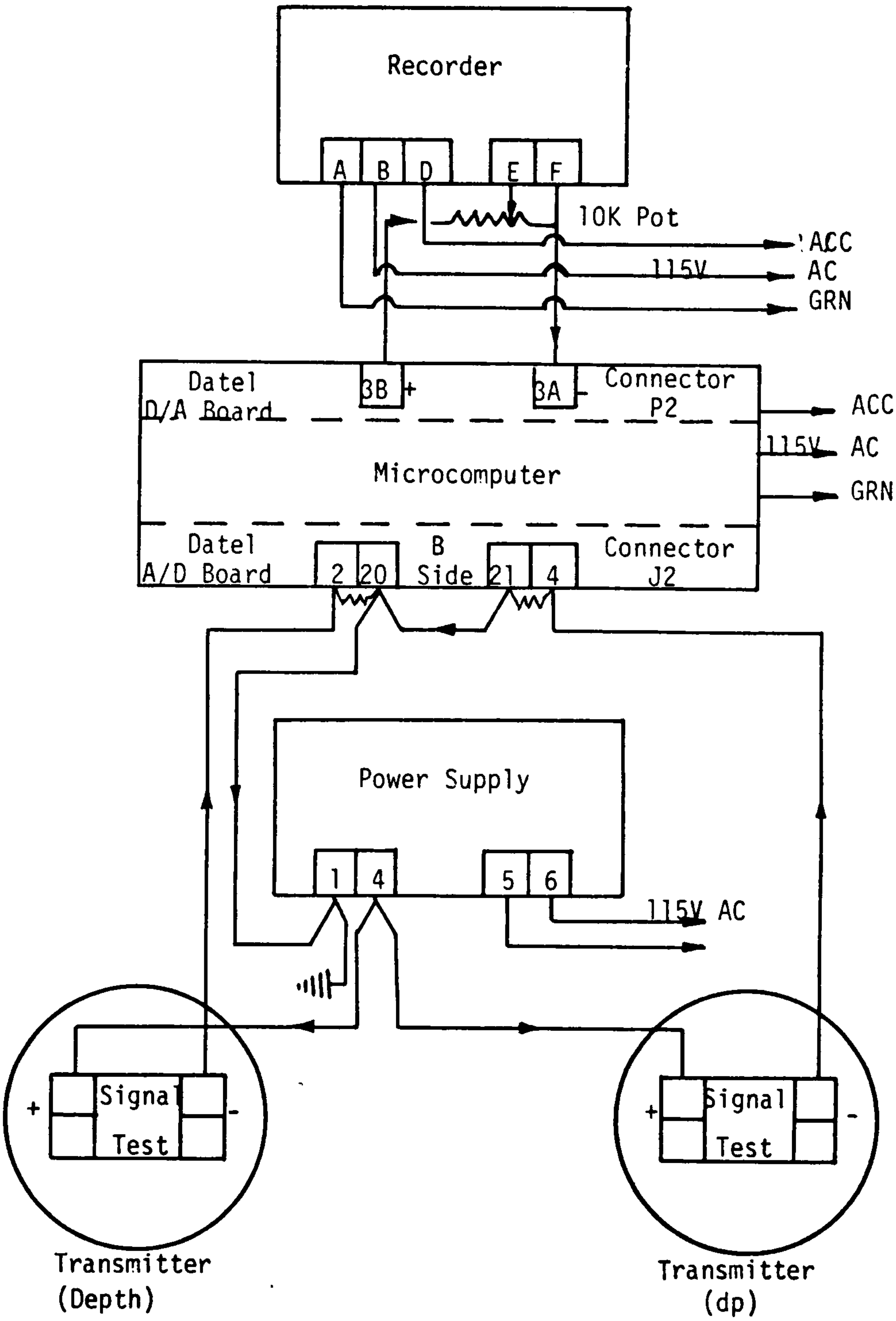


Figure 5. Interconnection Diagram

48 parallel I/O lines, 1K bytes of static Random Access Memory (RAM), sockets to accept 4K bytes of Read Only Memory (ROM/PROM) and a system clock. A block diagram of the BLC-80/10 is shown in Figure 6. A block diagram of the CPU is shown in Figure 7. The INS8080A N-MOS Large-Scale Integration (LSI) microprocessor is the central processor for the BLC-80/10. The microprocessor contains six general purpose 8-bit registers which can be used singly or in pairs, an 8-bit accumulator, a 16-bit stack pointer register and a 16-bit program counter. The 16-bit stack pointer allows the user's stack to be located anywhere within the memory space. This allows unlimited subroutine nesting levels and the capability of maintaining multiple stack areas. The 4K bytes of ROM can be installed in 1K increments. The INS8080A has 111 instructions which may be 8, 16, or 24 bits. The logic diagrams of the BLC-80/10 are included in Appendix A.

Converters. The Datel Systems Model ST-800 A/D converter accepts up to 32 single ended or 16 differential analog channels. [8] The converter can be operated in the program interrupt mode, under strictly program control or under direct memory address (DMA) control. When under program control from an 8080 type microprocessor it will select a particular channel, digitize the input and present the digital data to the computer for processing. The resolution of the digitized signal is 12 bits. Under program control, status (when a conversion is completed) is easily determined by reading the condition of bits in the status register. The ST 800 A/D system block diagram is seen in Figure 8.

The data acquisition section of the Datel ST 800 A/D system consists of an analog multiplexer, a buffer differential amplifier, a sample and hold amplifier and an A/D converter. The multiplexer selects

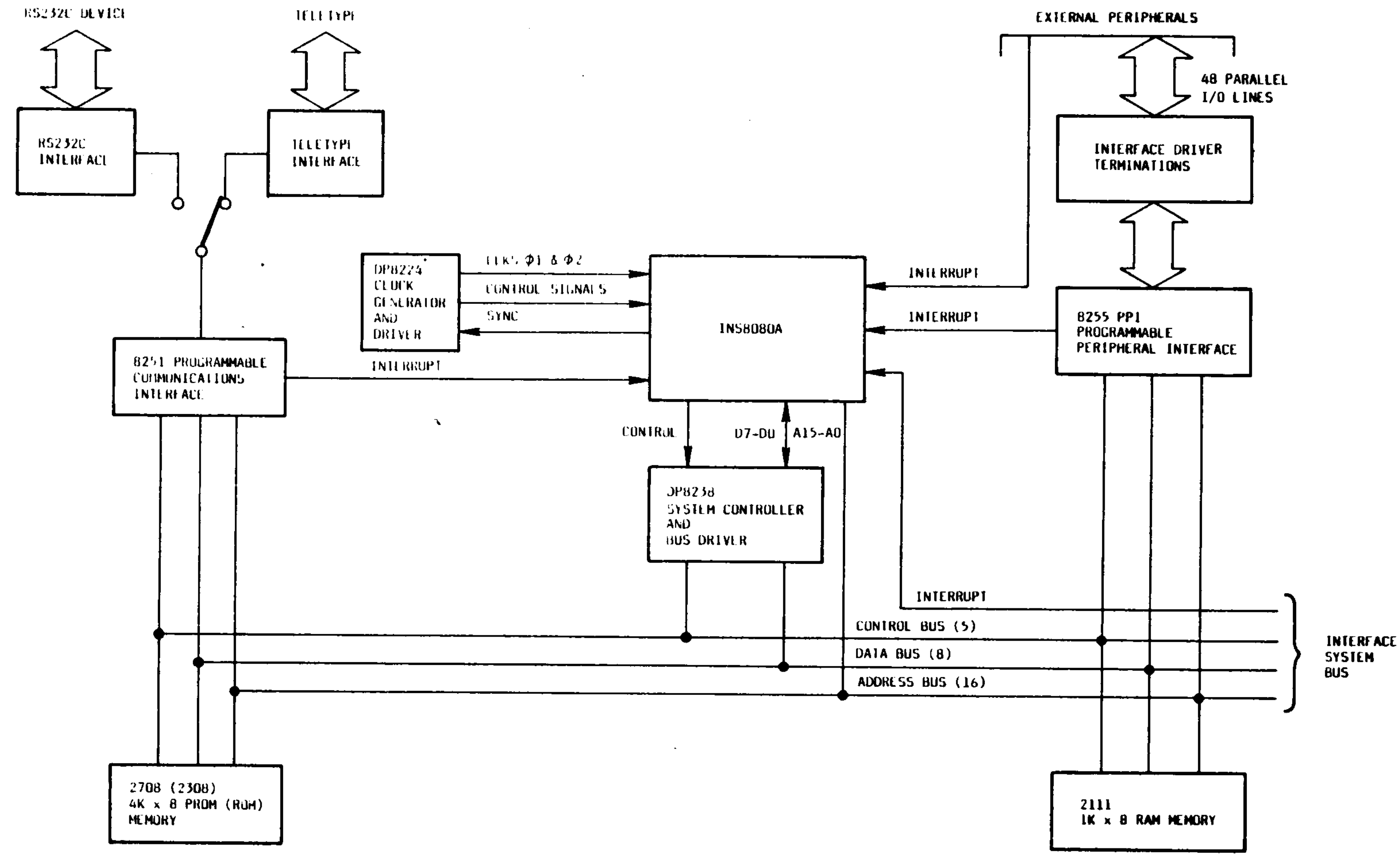


Figure 6. Board Level Computer - Block Diagram

Source: Series/80 RMC 80/10 Rack Mounted Computer Hardware Reference Manual. BLC 80/10 Board Level Computer Hardware Reference Manual, Pub. No. 420-305373-001A, May 1977, National Semiconductor Corp., Santa Clara, Calif.

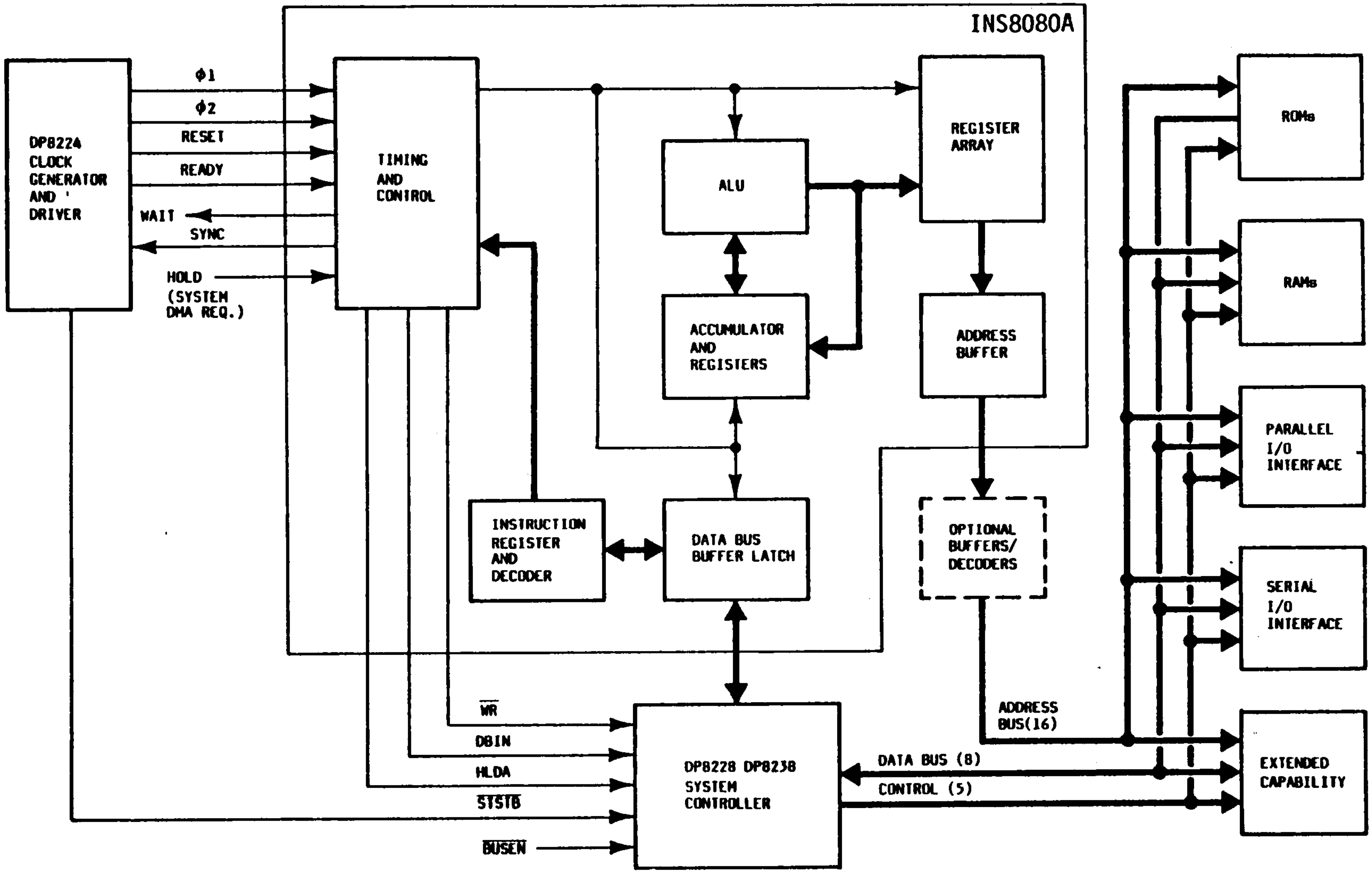


Figure 7. CPU Functional Block Diagram

Source: BLC 80/10 Board Level Computer Hardware Reference Manual, Pub. No. 420-305373-001A, May 1977, National Semiconductor Corp., Santa Clara, Calif.

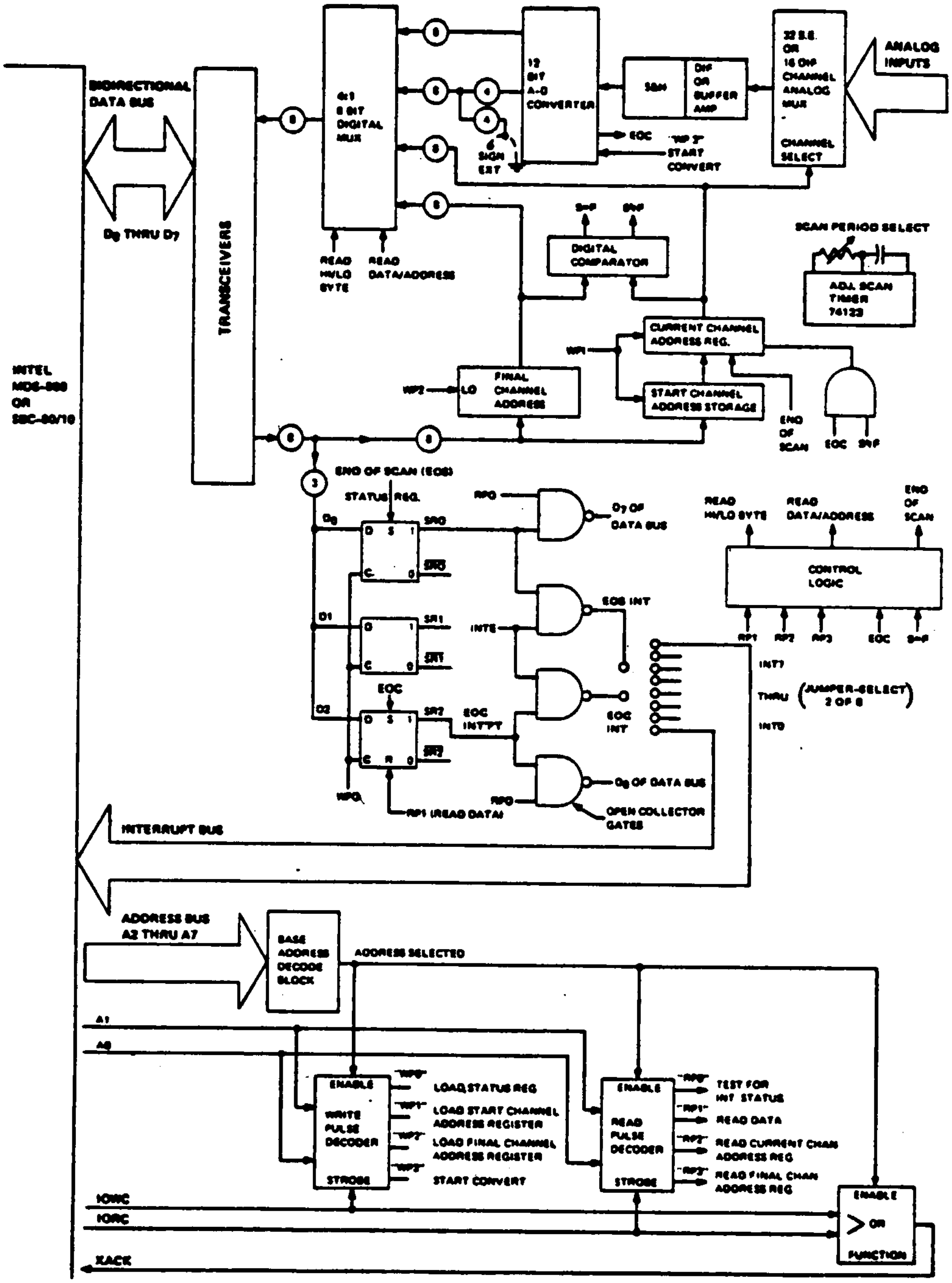


Figure 8. ST 800 A/D System Block Diagram

Source: Sine Trac Series Model ST-800-325/16D A/D Peripheral Systems Instruction Manual.

any of the 16 differential channels under the control of the number in the current channel address register. The buffer differential amplifier provides high input impedance and provides a means of converting differential channel inputs to single ended outputs. The sample and hold amplifier acquires the signal and holds it during the conversion since the output of this section goes to the A/D converter. The acquisition time is on the order of 12 microseconds. The A/D converter employed in this system uses a successive approximation type of conversion. The conversion time is 8 microseconds. The typical data transfer time for the A/D Board is 36 microseconds when under program control. This is the minimum time between samples that the converter can obtain.

The Datel Systems Model ST-800-DA4 D/A converter accepts up to 4 12-bit digital signals. [9] This system can be operated under program control or in the program interrupt mode. When under program control from an 8080 type microprocessor it will accept digital data from the CPU, select a particular channel, convert the data and output to an analog signal. The data is converted by utilizing three quad current switches, two thin film resistor networks, a precision zener reference circuit and an output amplifier. In this system the output is a 0 to 5V dc signal. Status of the conversion is again determined by reading bits in the status register. The ST-800 DA4 D/A system block diagram is shown in Figure 9.

The microprocessor system programming is considered in the following chapter.

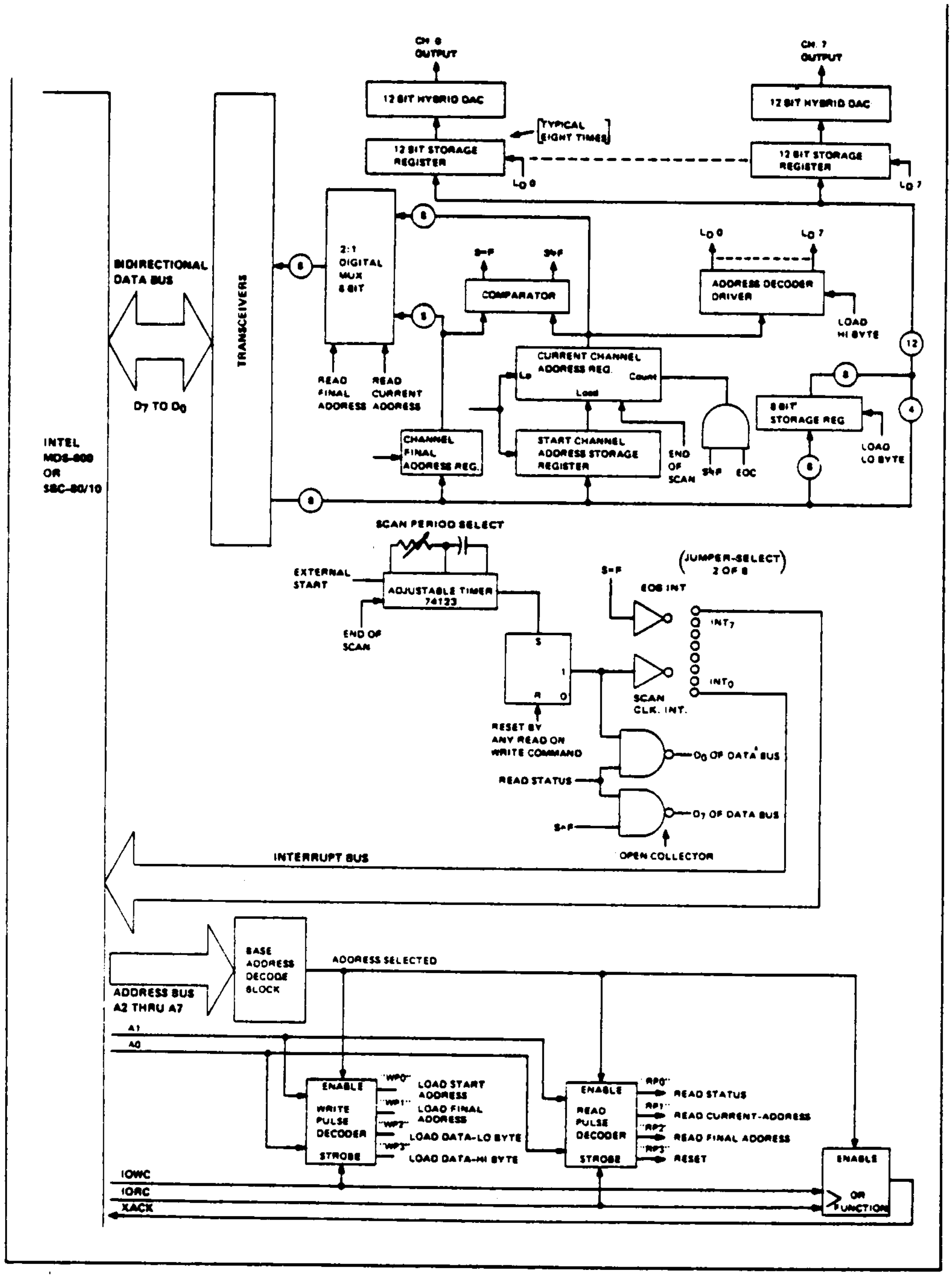


Figure 9. ST 800 D/A System Block Diagram

Source: Sine Trac Model ST-800DA Series D/A Peripheral Systems Instruction Manual.

CHAPTER IV

SOFTWARE CONSIDERATIONS

Code

All instructions were programmed in 8080 Assembly language. [10] The source code was edited, assembled or converted to object code, and programmed into two 2708 1K PROMS by using the operating system in a microcomputer development system. [11] The PROMS were then installed in the microcomputer used in the flowmeter. Use of the development system greatly simplified programming by allowing all programming to be done in assembly language instead of machine code which would have to be hand loaded. The programming could have been further simplified by using Fortran 80, also available on this development system. However, considerable more memory space would be required for this particular case. Since our system is limited to 4K ROM and 1K RAM, programming in Fortran 80 would have required more memory than we had available.

Programming

The general flowchart is shown in Figure 10. In the first step, a block of ROM memory was chosen and all constants, including the pipe radius R and the flow constant K, were loaded into this block. ROM was chosen for storage because there is four times as much ROM memory as RAM. The constant, $\pi R^2/2$ is calculated for the pipe to be used and stored in RAM memory by the microcomputer. This constant can then be recalled when needed for the flow calculation.

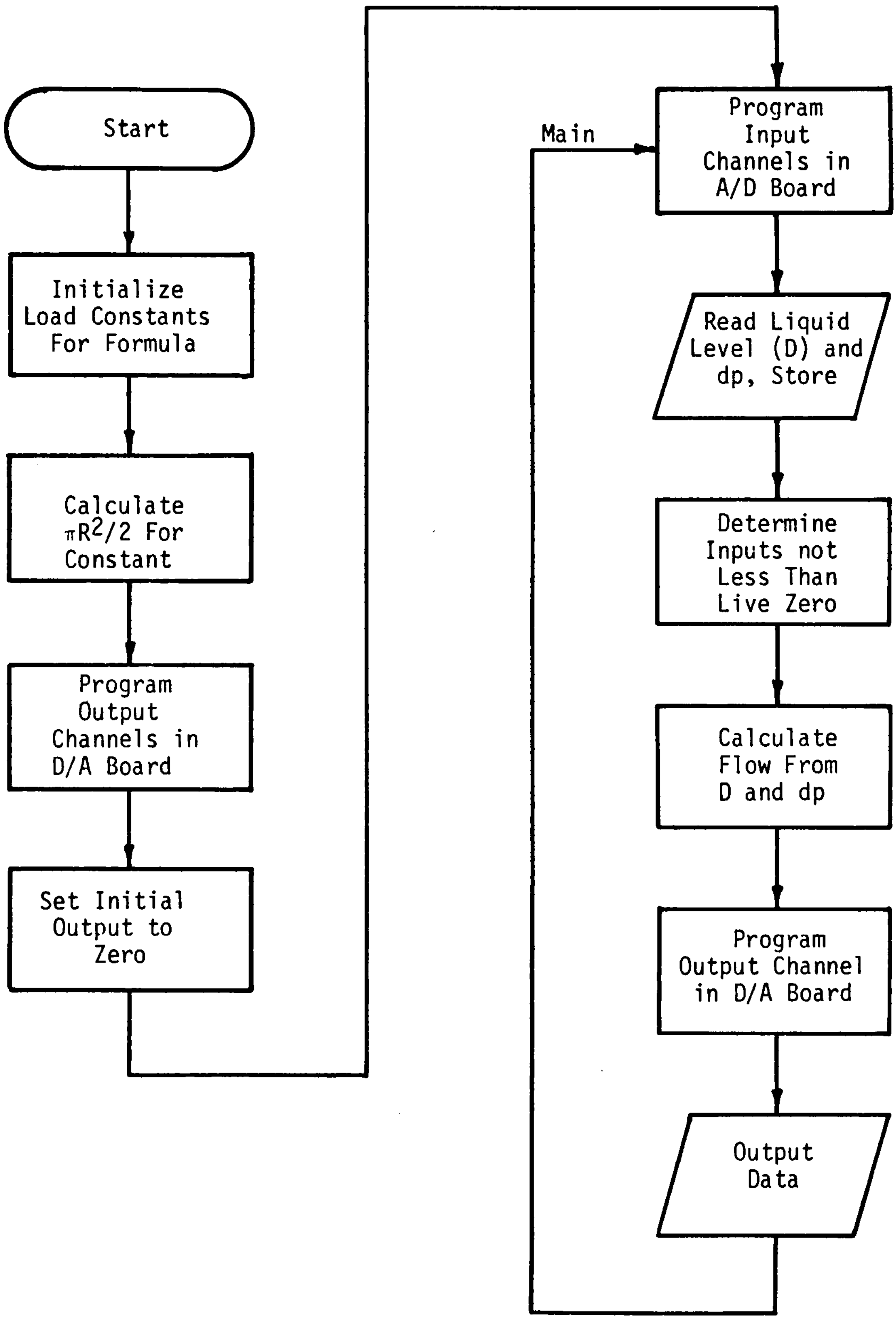


Figure 10. System Flow Chart

Of the four output channels available on the D/A converter, only one output is required. The address of this channel must be loaded into the converter and its initial output set equal to zero. [9] All the initial conditions are now established.

Two of the 16 differential inputs available on the A/D converter Board are used. The starting channel and final channel addresses must be programmed into the A/D Board. The inputs are read through these channels and stored by another series of programming steps.

The analog input signals coming from the liquid level and flow transmitter vary from 4 to 20 milliamps. Four milliamps represents zero flow velocity and zero liquid depth. It is not uncommon for a transmitter's output signal to be slightly less than 4 milliamps when the transmitter input is zero. Each A/D converter output is checked to ensure that it is not less than the digital input resulting from a 4 milliamp input. If the input is less than this value, it is changed to the value representing 4 milliamps and this is stored instead.

The total flow is then calculated for these values of liquid level and flow velocity using the equation developed in Chapter II, "Description of the Process." All mathematical manipulations are performed by using subroutines FMATH and SQRT. FMATH, a reverse polish arithmetic processor, takes a string of variables, converts the variables to floating point notation, and then performs the arithmetic processes designated in the strings. It also converts floating point notation back to fixed point when required. SQRT finds the square root of specified variables. Using these subroutines and the polish strings greatly simplifies programming and requires minimal memory storage.

The D/A board is next programmed to output this value on the correct output channel. The program then returns to the data input step. The output value is held until the results of a new calculation are available.

The detailed flow charts are included in Appendix B. The resulting program listing is given in Appendix C.

CHAPTER V

RESULTS

Performance of the New System

Reliability. In lab tests, the new system proved considerably more reliable than the purchased flowmeter. The system operated for a four-month period without equipment failure of any type. During this period, the zero and full-scale outputs of the differential pressure transmitters drifted less than 0.25%. Recalibration of the transmitters was not required. The recorder was recalibrated on one occasion following an output drift of approximately 1%.

Accuracy. During this period, system accuracy was checked by applying simulated pressure signals to both differential pressure transmitters. These pressures were varied over the pressure range corresponding to flow velocities and liquid levels from 0% to 100% of maximum. Precise calculations were made determining the theoretical volume flow. This was compared to the recorder reading. The recorder had been calibrated for a 0 to 5 volt range, corresponding to a flow range of 0 to 29.69 cubic ft/min.

A graph of the deviations from the theoretical output as a percent of full-scale output is shown in Figure 11 for all data. For all tests the maximum deviation as a percent of full-scale reading was 9.6%. In most cases, the deviations are less than ± 5%. The cases in which the deviations exceed these values represent liquid levels greater than 95% or less than 5% maximum. The deviation was 0% at 0 flow. So the experimental accuracy is demonstrated to be well within the required 15% by this graph.

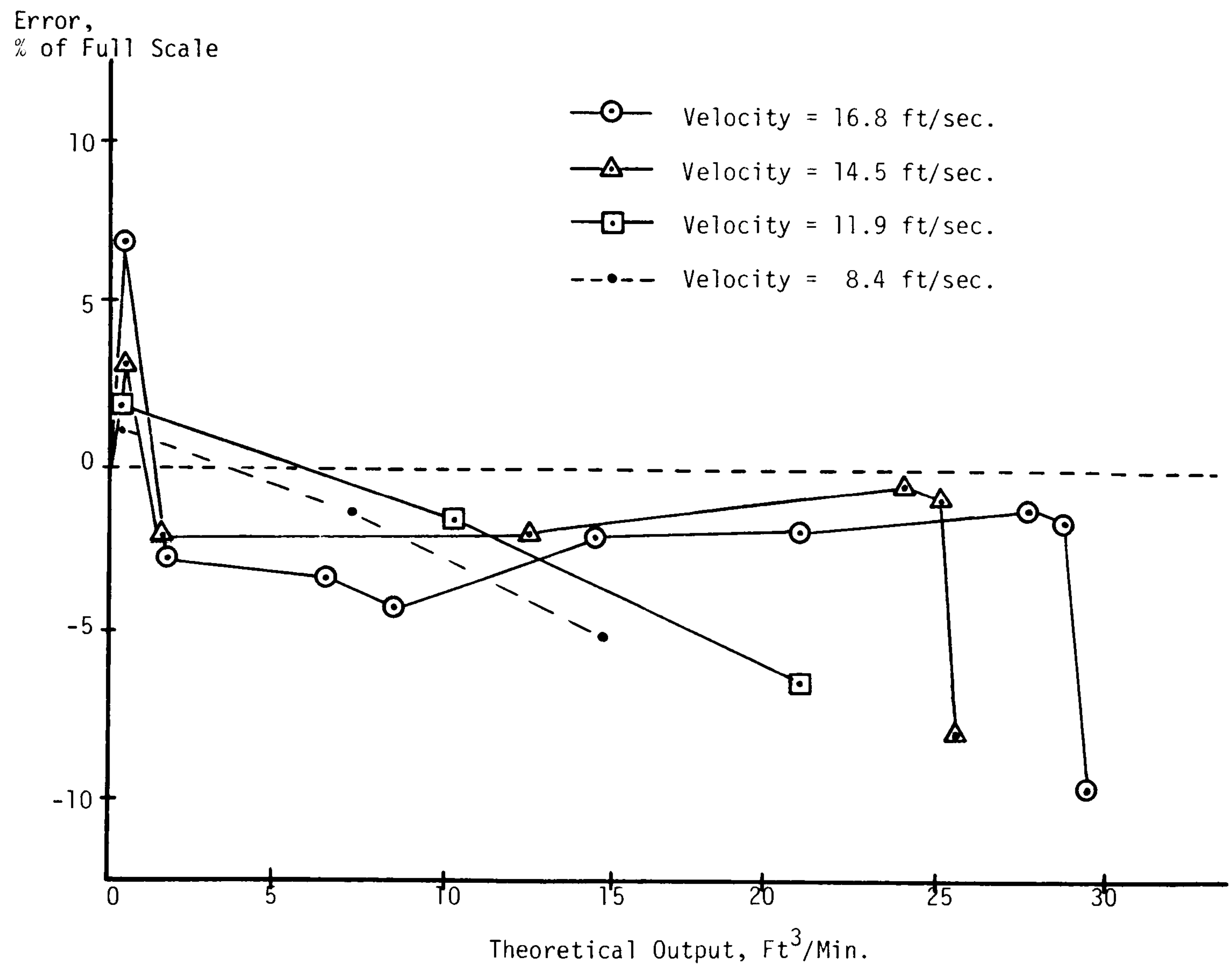


Figure 11. Output Error as a Percent of Full-Scale Output

Time Response. No calculations were made to determine the theoretical speed of response to an input change. However, the system output responded to input changes faster than the recorder could follow. The recorder has a full-scale response of 1.5 seconds or less, therefore, the system did meet the requirement for consecutive flow measurement outputs to occur in less than one minute.

Flexibility. During the test, the system was reprogrammed for installation at another site with a different pipe size and a different maximum flow velocity. This change was made in less than four hours by using the development system. The system is therefore very flexible and meets this last system requirement.

Cost Comparisons

Total typical hardware cost for the flowmeter developed on this project is $4719.17. A breakdown of the component cost is given in Table 1. The price of the commercial flowmeter was $6500 in 1978 or approximately $1800 more than the hardware cost for our unit. Development costs for one flowmeter exceeded this $1800 difference. However, installation of three or more flowmeters would bring the total cost per unit to less than the cost of the commercial unit when the development costs are prorated.

Total costs per unit can be reduced again when two or more flowmeters are installed in the same general vicinity. The microcomputer can process inputs from a maximum of four different liquid streams using the present A/D and D/A Board Combination. Moderate programming changes would be required to do this. Hardware costs would be only $2950.17 for each of the three additional flow measurements.

TABLE 1

COST OF TYPICAL FLOWMETER SYSTEM HARDWARE

Air Pump 3 @ $26.99	$ 80.97
A/D Converter	650.00
D/A Converter	559.00
8" Annubar Flow Sensor	400.00
Microcomputer	560.00
2708 PROM chips 2 @ $23.10	46.20
Power Supply	129.00
Differential Pressure Transmitter 2 @ $764.50	1529.00
Electronic Single Pen Recorder	765.00
Total	$4719.17

When everything is considered, this system should be less expensive than the commercial unit.

CHAPTER VI

SUMMARY AND CONCLUSIONS

A flowmeter was designed and developed by the author to measure total volume flow in partially full pipes because a reliable, accurate flowmeter was not commercially available. The system was designed to use an Annubar primary sensor with a differential pressure transmitter to obtain flow velocity information and a bubbler tube with a differential pressure transmitter to obtain liquid level information. Equations were derived by the author that related this information to total volume flow. A microcomputer was included in the system to perform the flow calculations. The microcomputer was programmed by the author with the assistance of Mr. Bill Osborne. A strip chart recorder was added to the system to provide a record of the total flow.

The system exceeded all performance requirements. Down time for maintenance, system accuracy and response time are all within desired performance limits. In addition this system is much more flexible than the previous instrument systems. With the increased emphasis placed on monitoring liquid discharge in the chemical industry, this flowmeter should fill a need that can not presently be met by commercial systems.

Even though accuracy is within specifications, system accuracy can be improved. A look-up table can be included in ROM so that values for pipe areas would be available whenever the liquid level exceeds 90% maximum or is less than 10% maximum. This would enable the output to be much more accurate and would allow the number of table entries to be kept to less than twenty. This possibility would be investigated if the system were redone.

REFERENCES

REFERENCES

1. Samuel L. Selby (ed.), CRC Standard Mathematical Tables, 14th ed. Cleveland, Ohio: Chemical Rubber Company. Pp. 409, 490-492. (1965)

2. Annubar Primary Flow Sensors for Gas, Liquid or Steam, Publication E-100. Boulder, Colorado: Dieterich Standard Corp. Pp. 14, 17-18. (1974)

3. N. O. Thoi and W. K. Soh, "On the Averaging Technique and the Discharge Coefficient of the Annular-Averaging Flow Sensor," I.S.A. Transactions, Vol. 18, No. 1, pp. 41, 46.

4. F. W. Sears and M. W. Zemansky, University Physics, 3rd ed. Reading, Mass.: Addison-Wesley Publishing Co. Pp. 316-317. (1964)

5. Model 1151DP Alphaline Differential Pressure Transmitter, Data Sheet #2104. Minneapolis, Minn.: Rosemount, Inc. (1971)

6. BLC 80/10 Board Level Computer Hardware Reference Manual, Publication No. 420305373-001A. Santa Clara, California: National Semiconductor Corp. (1977)

7. National Semiconductor Series/80 Databook. Santa Clara, California: Semiconductor Corp. (1979)

8. Sine Trac Series Model ST-800-32S/16D A/D Peripheral Systems Instruction Manual, Document No. ST8AMH3701. Canton, Mass.: Datel Systems, Inc. (1978)

9. Sine Trac Model ST-800DA Series D/A Peripheral Systems Instruction Manual, Document No. DA8-DH1611. Canton, Mass.: Datel Systems, Inc. (1976)

10. Intel 8080/8085 Assembly Language Programming, Manual No. 9800 940. Santa Clara, California: Intel Corp. (1979)

11. MCS-80 Users Manual, No. 98-1153D. Santa Clara, California: Intel Corp. (1977)

APPENDICES

APPENDIX A

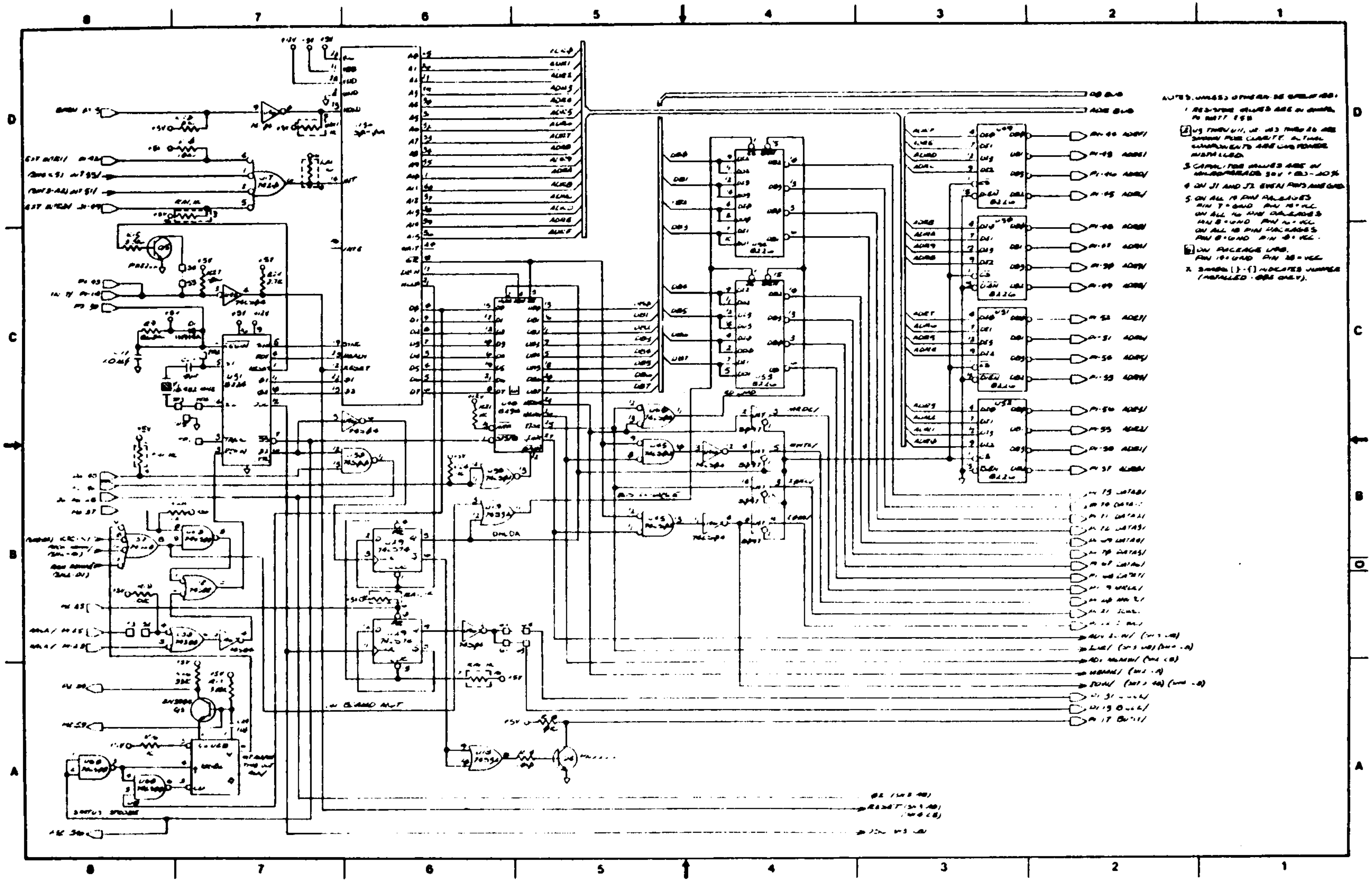


Figure A-1. BLC 80/10 Logic Diagram

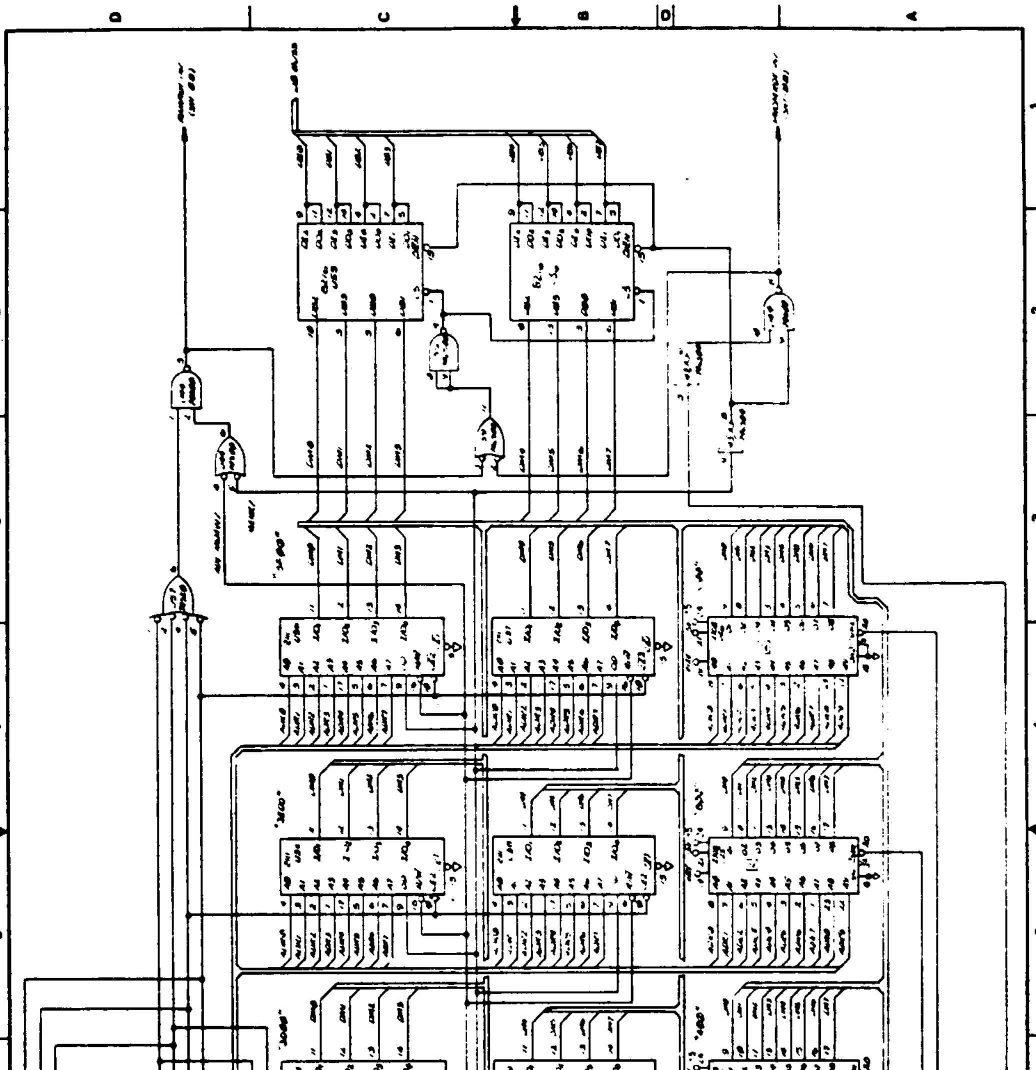


Figure A-1 (continued)

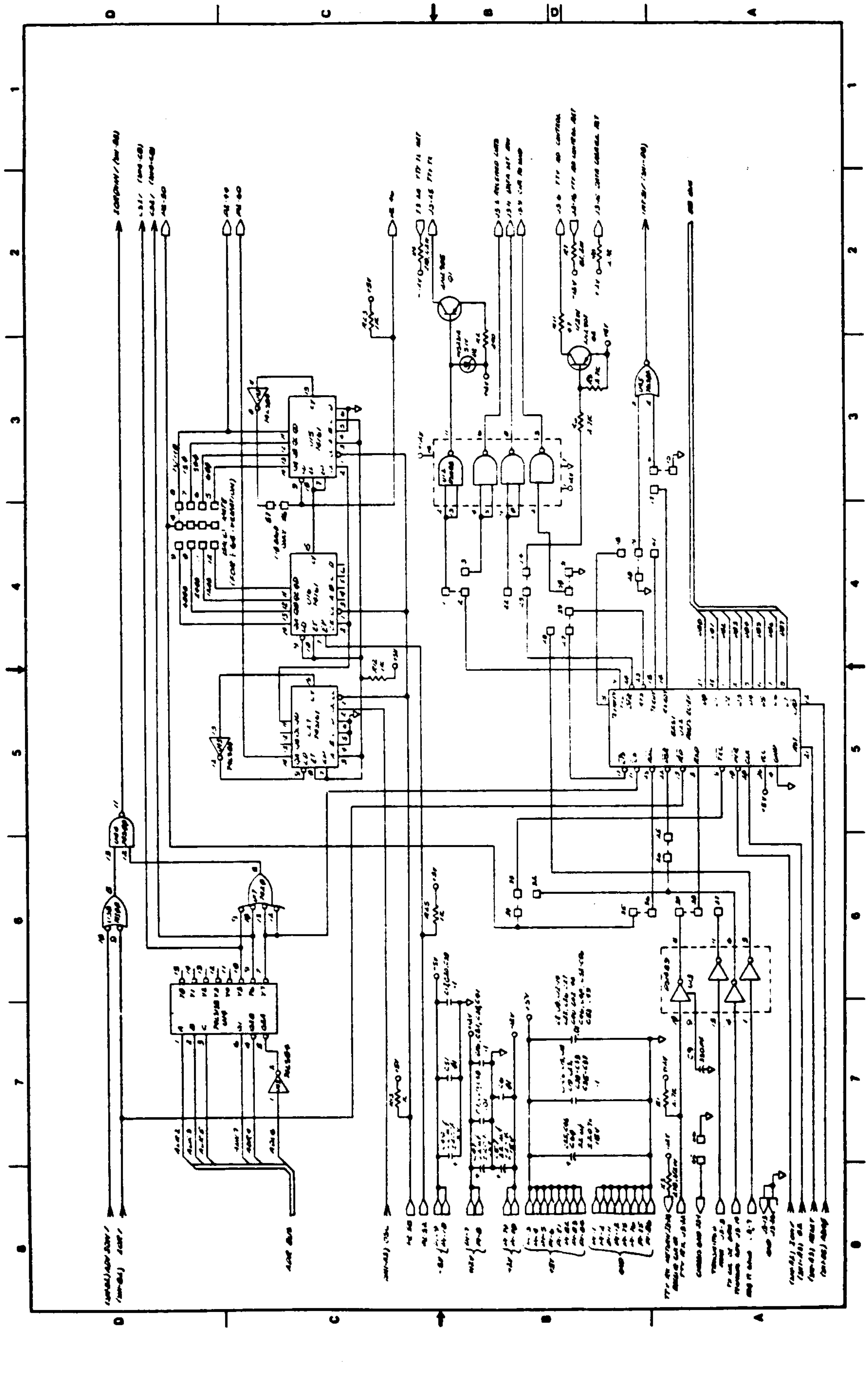


Figure A-1 (continued)

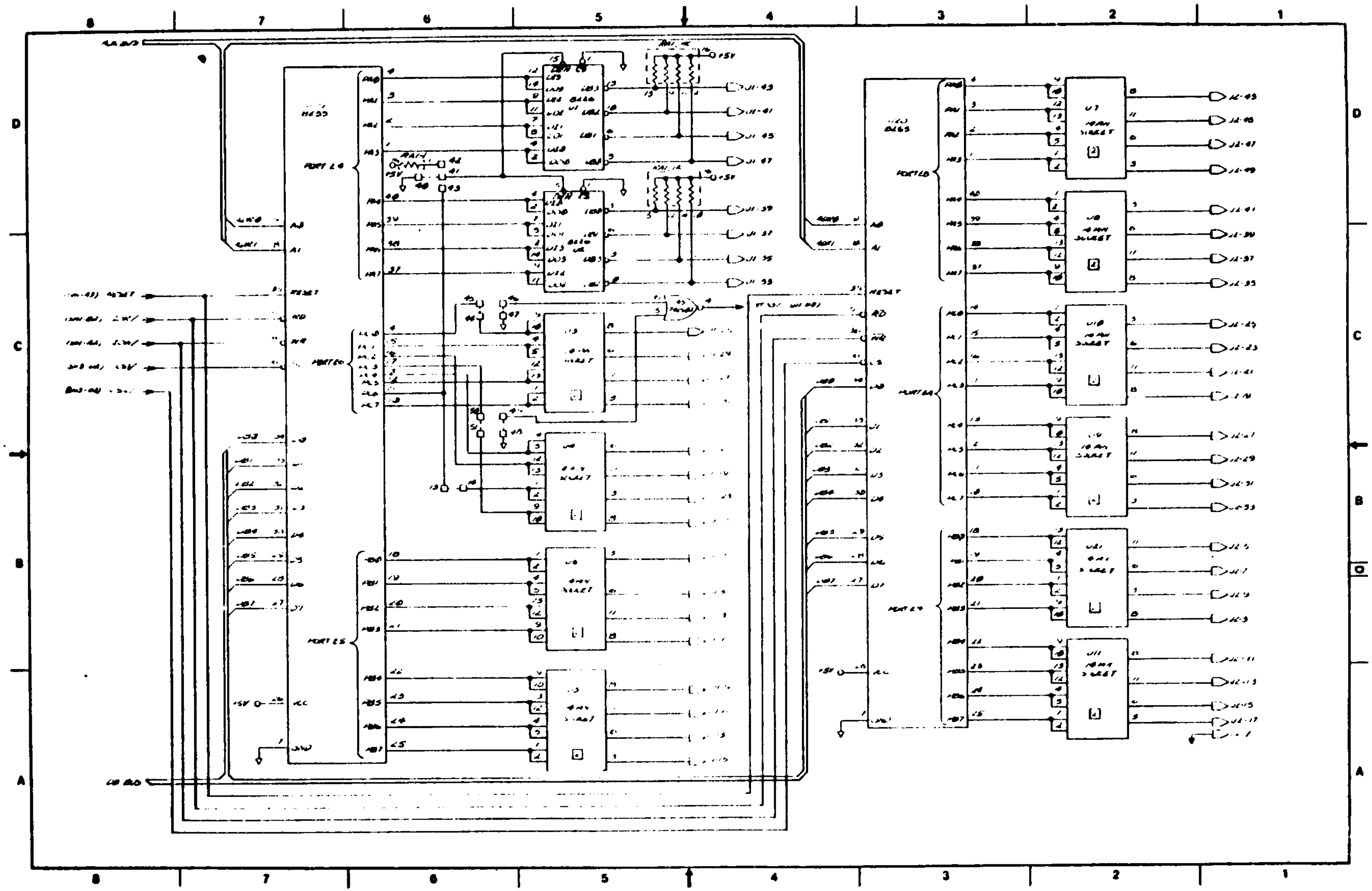


Figure A-1 (continued)

Source: BLC 80/10 Board Level Computer Hardware Reference Manual.

APPENDIX B

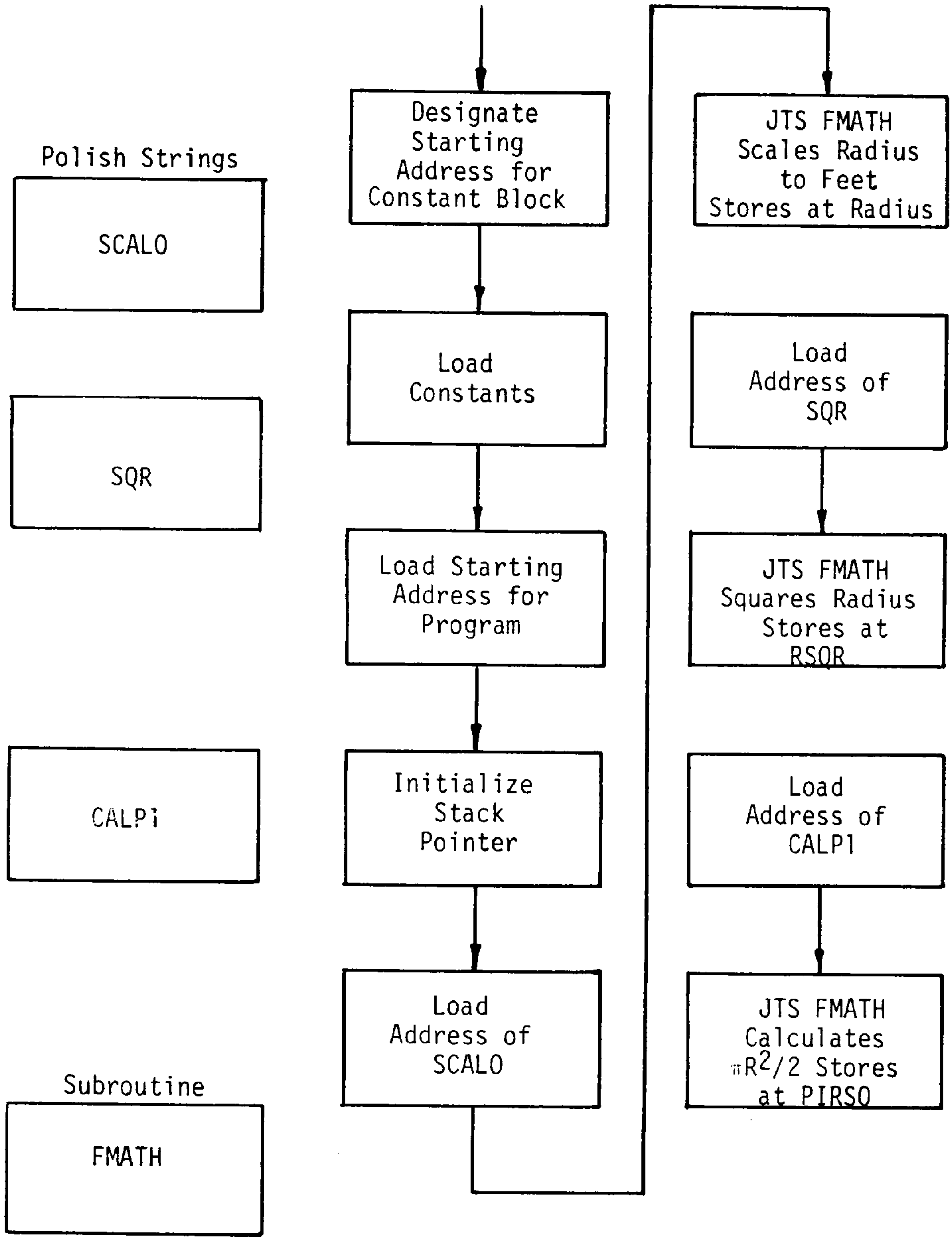


Figure B-1. Flow Chart for Loading and Calculating Constants

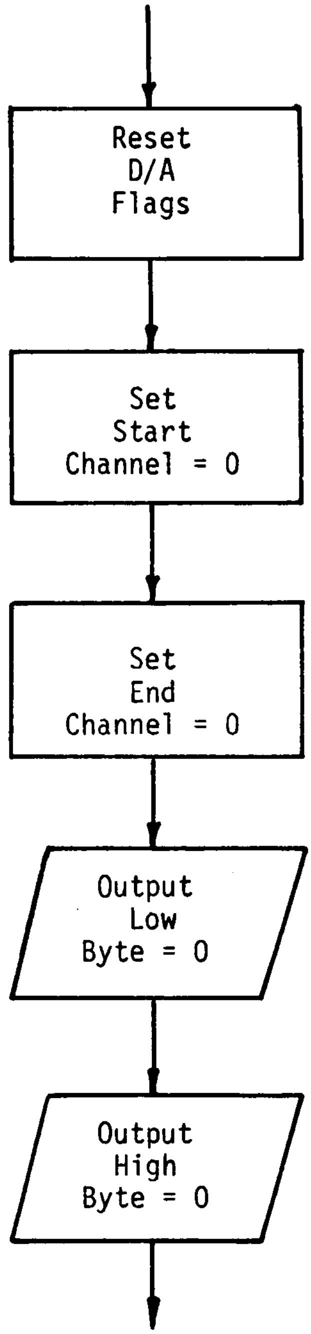


Figure B-2. Flow Chart to Initialize and Reset D/A System

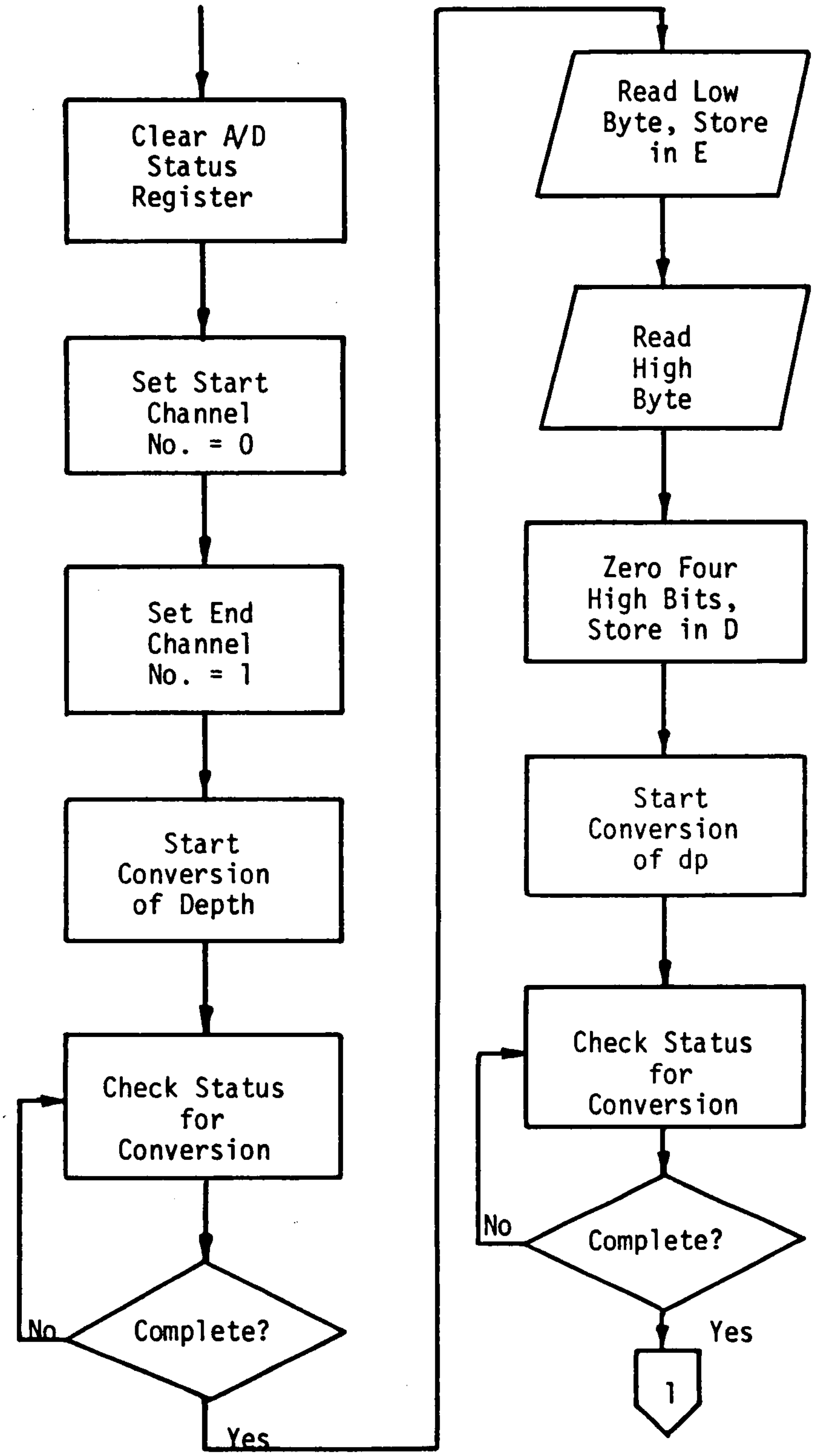


Figure B-3. Flow Chart for Programmed A/D Input Routine

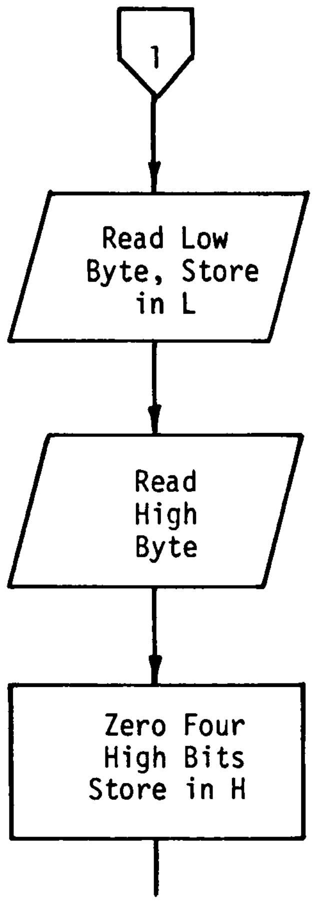


Figure B-3. (continued)

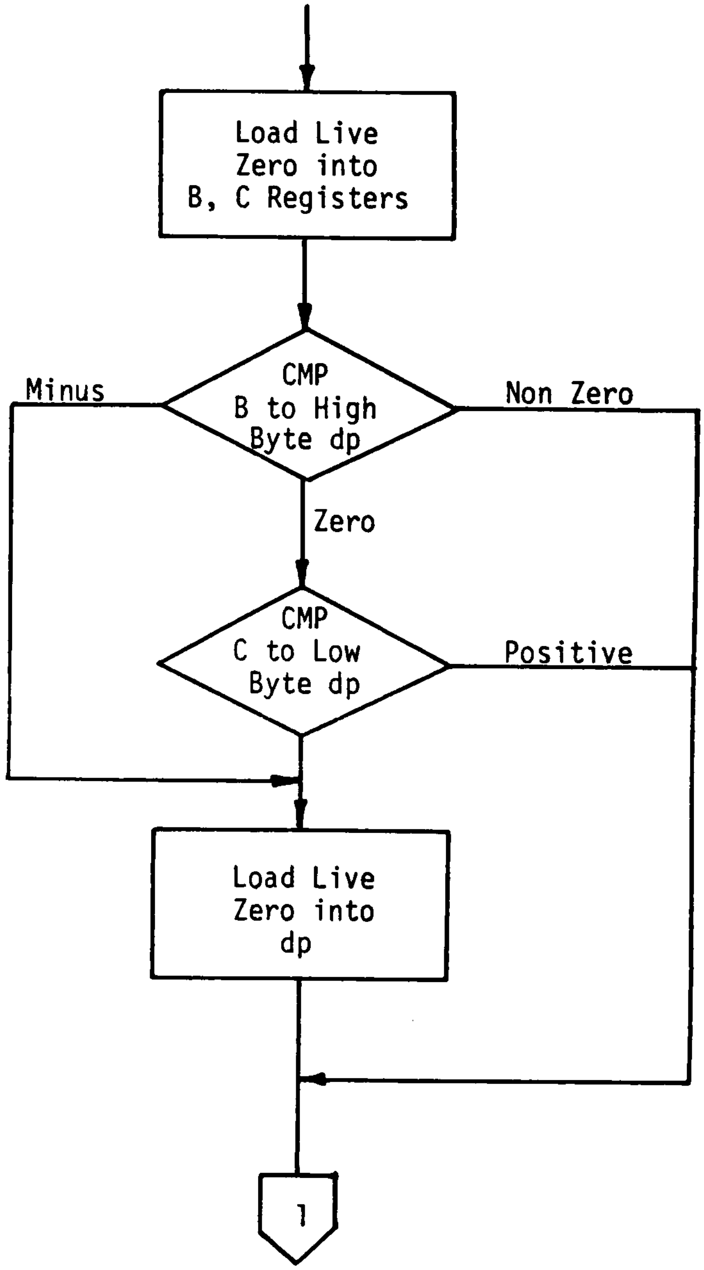


Figure B-4. Flow Chart for Ensuring Inputs are Equal to or Greater than Zero

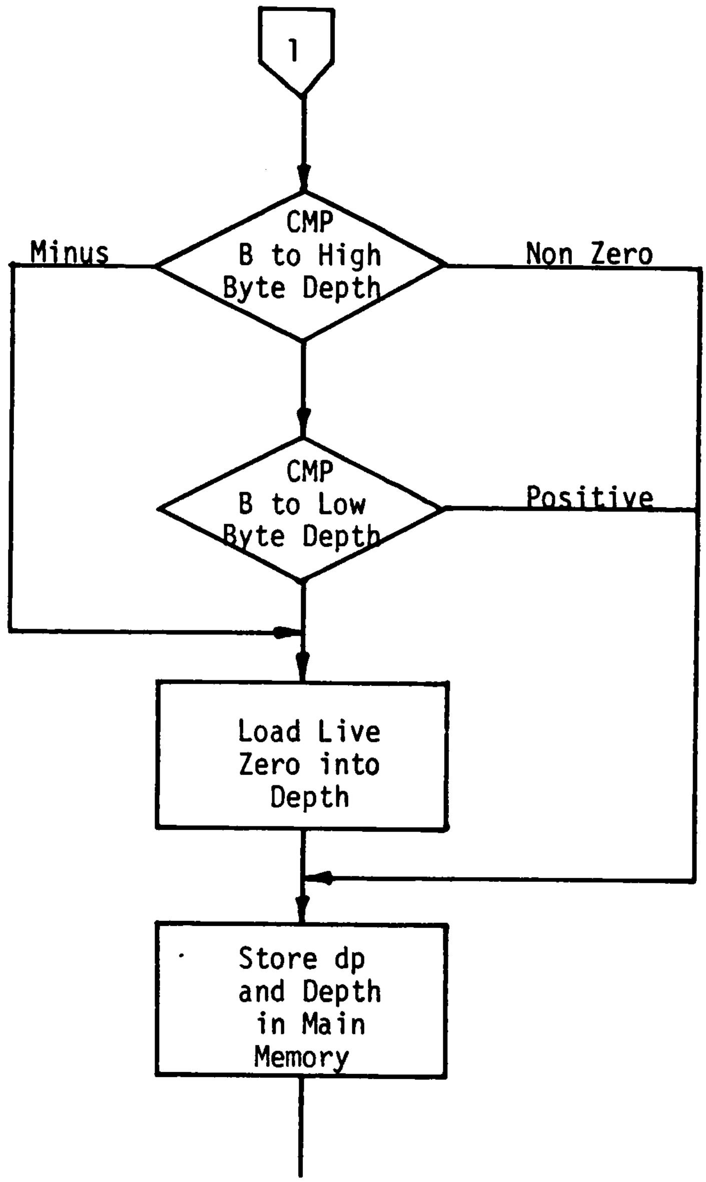


Figure B-4. (continued)

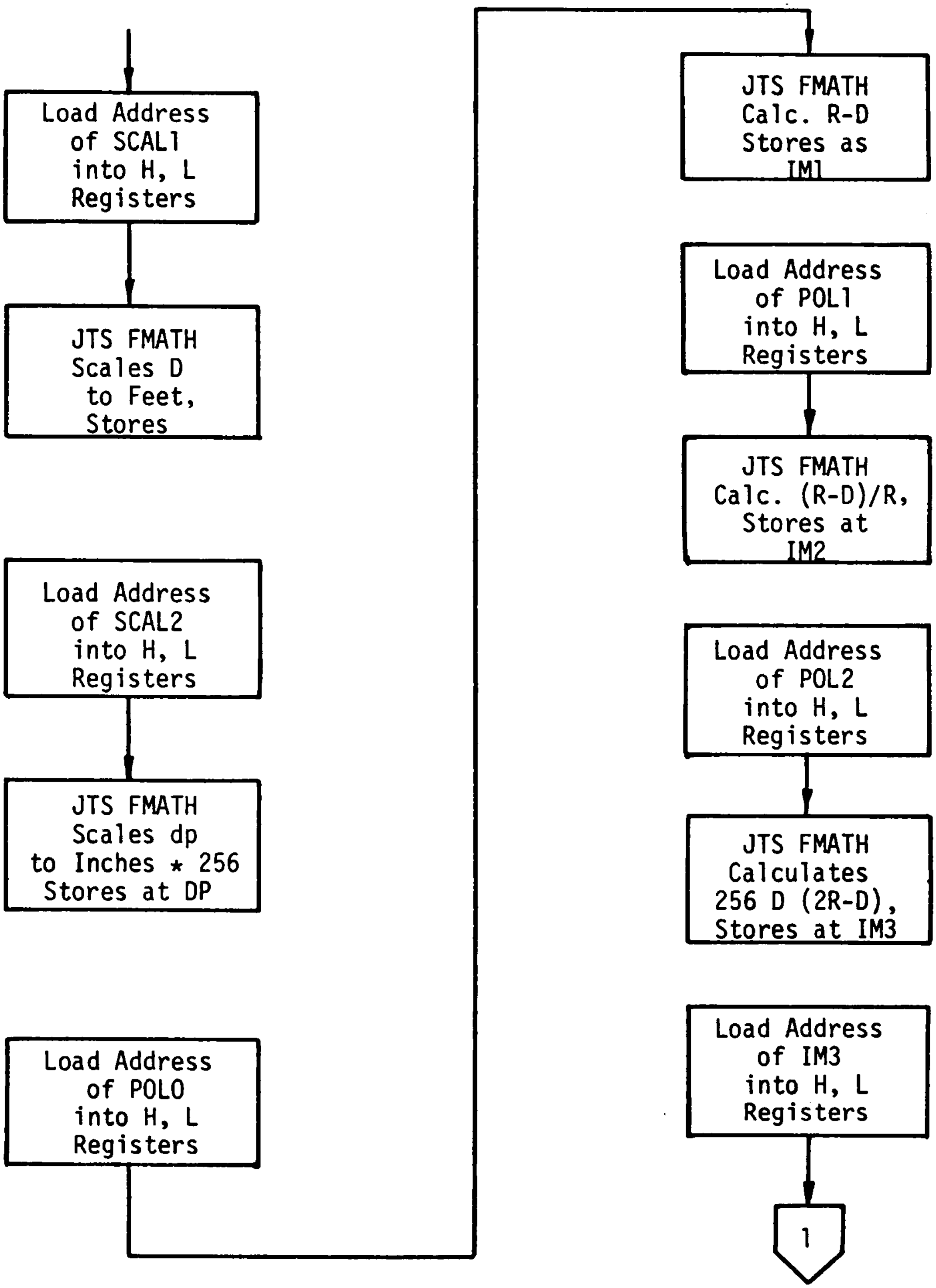


Figure B-5. Flow Chart for Flow Calculation

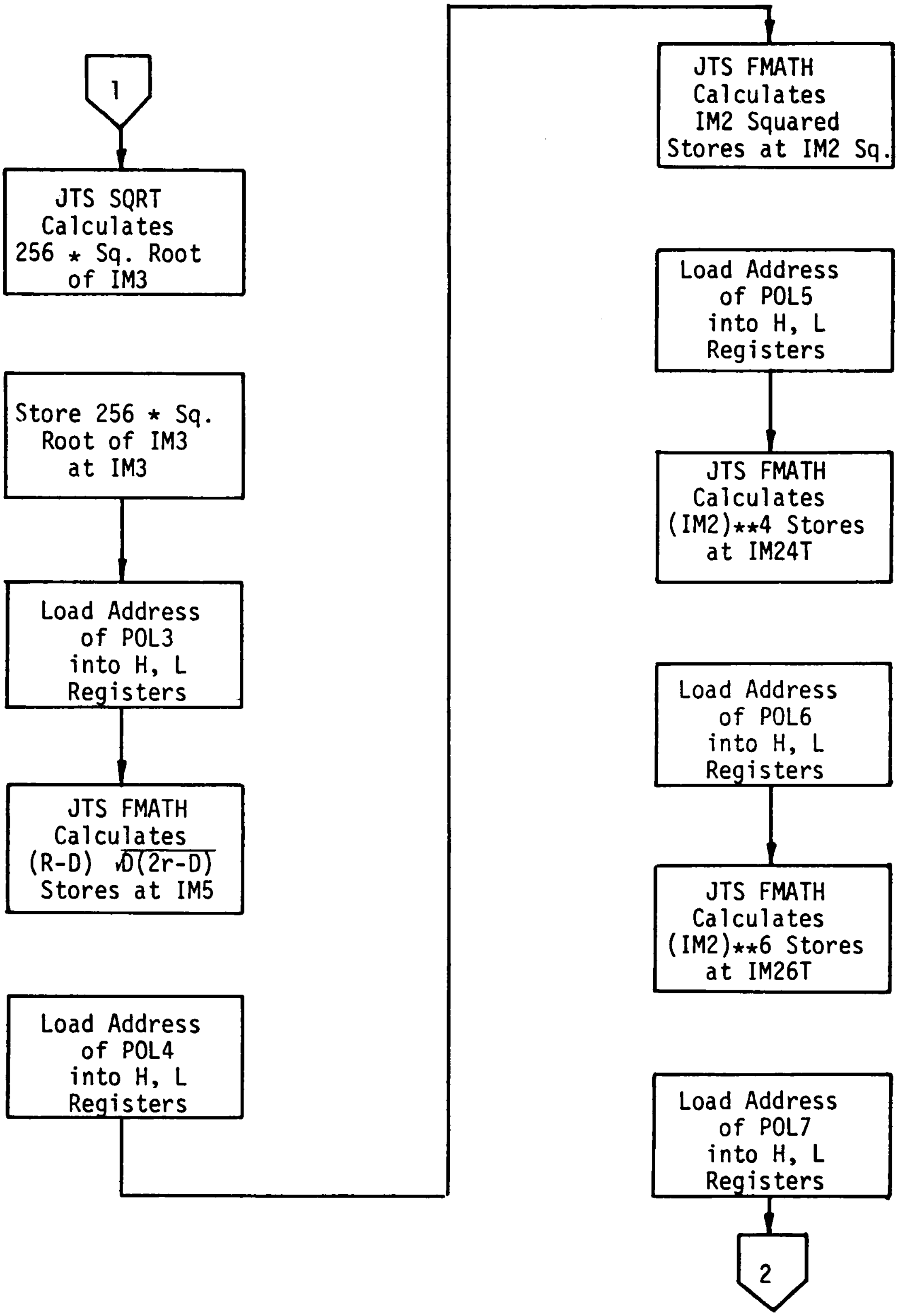


Figure B-5. (continued)

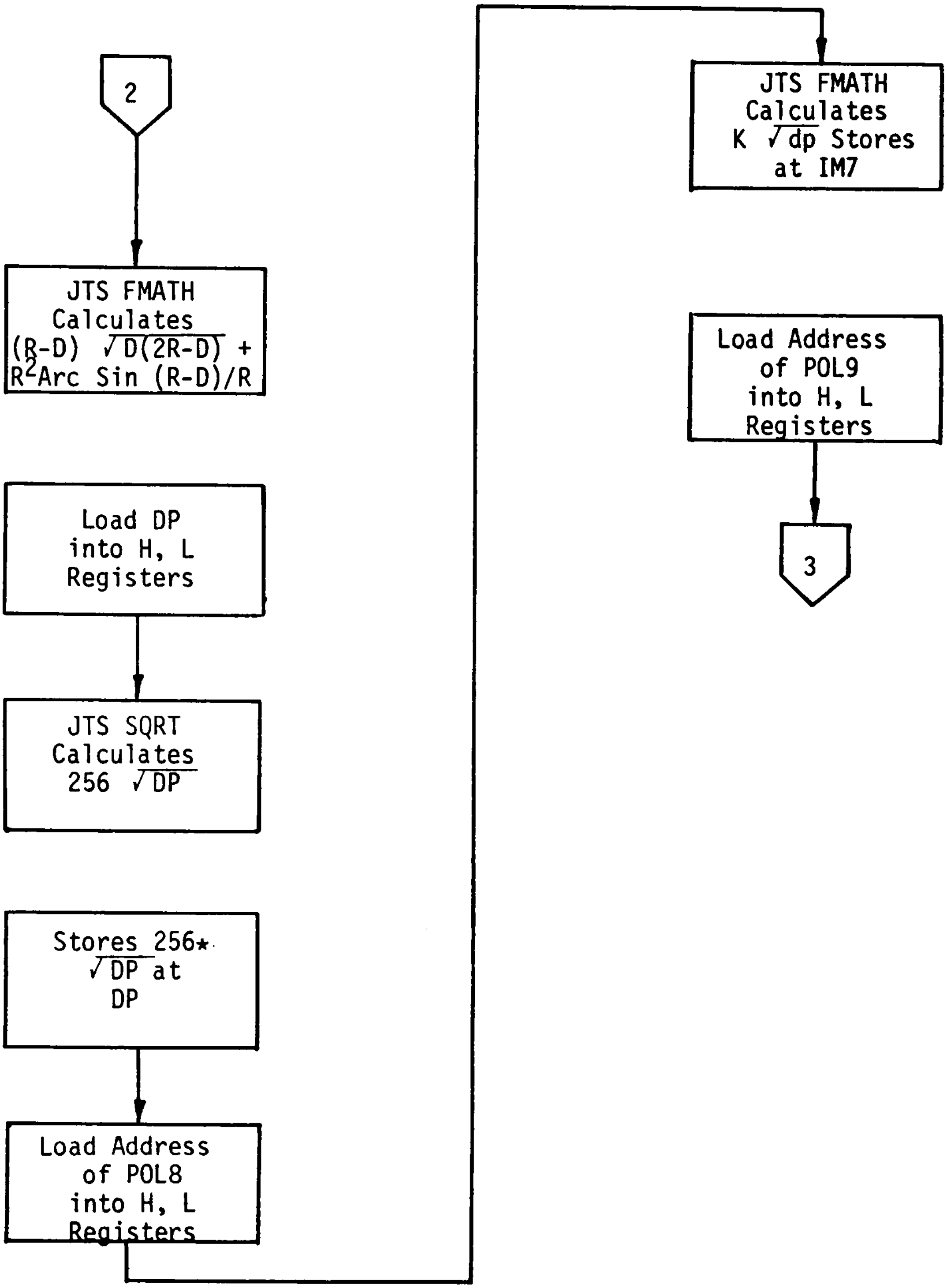


Figure B-5. (continued)

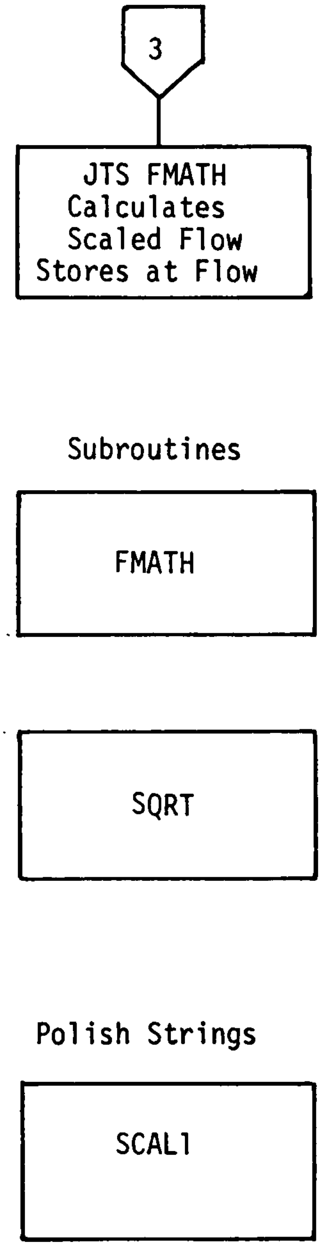


Figure B-5. (continued)

Polish Strings (continued)

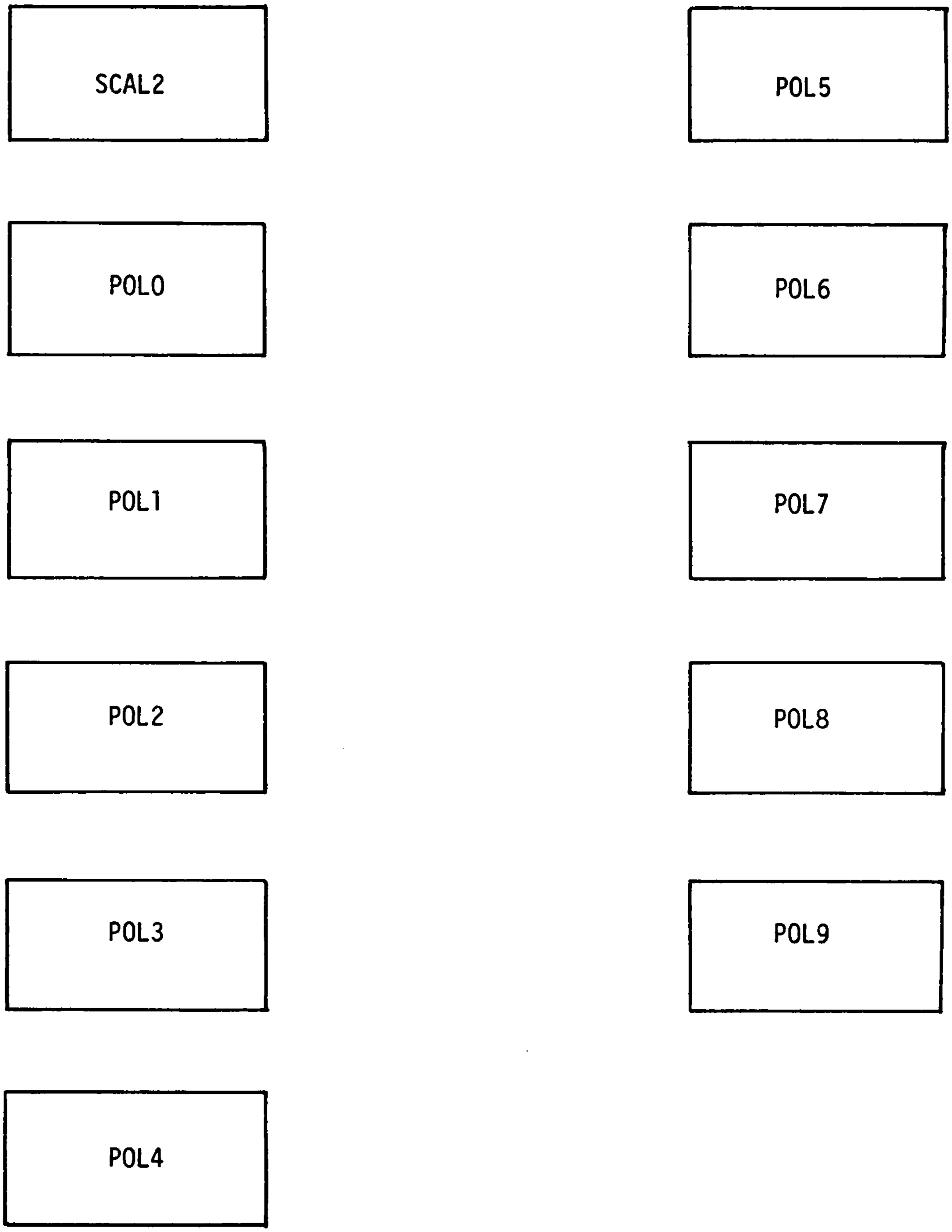


Figure B-5. (continued)

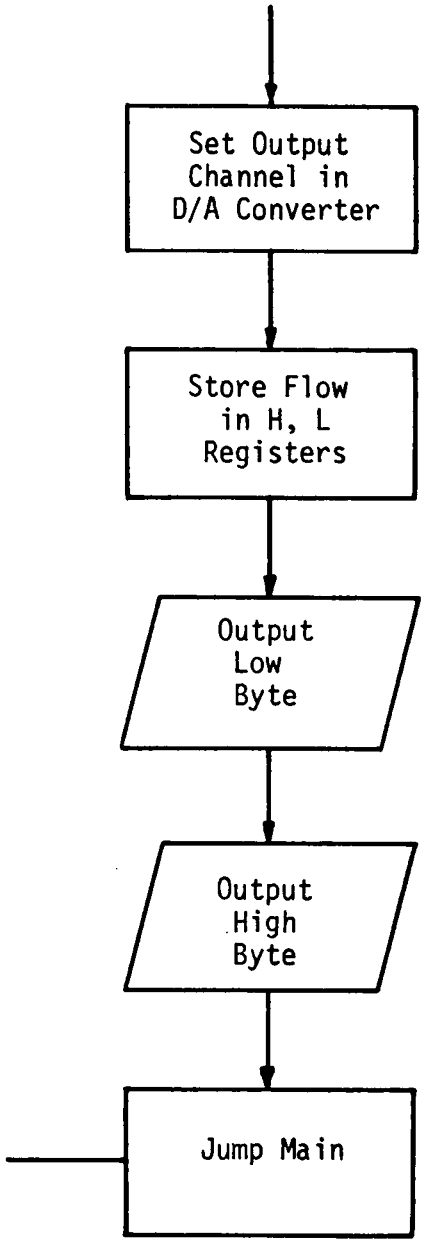


Figure B-6. Flow Chart for Programmed D/A Output Routine

APPENDIX C

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ       SEQ          SOURCE STATEMENT

0700             1          ORG      700H
0700 B87A        2 PI:      DW       31416     ;DIVIDE BY 10000 FOR PI
0702 4B00        3 RAD:     DW       75        ;EQUAL TO RADIUS IN FT*100
0704 9006        4 K:       DW       1680      ;CONSTANT TIMES 1000
0706 3303        5 ZERO:    DW       819       ;LIVE PROCESS ZERO
0708 FF0F        6 FULLSC:  DW       4095      ;PROCESS FULL SCALE
070A 204E        7 X20000:  DW       20000
070C 0300        8 X3:      DW       3
070E 0200        9 X2:      DW       2
0710 0001       10 X256:    DW       256
0712 0F00       11 X15:     DW       15
0714 5001       12 X336:    DW       336
0716 2800       13 X40:     DW       40
0718 0600       14 X6:      DW       6
071A 0100       15 X1:      DW       1
071C E803       16 X1000:   DW       1000
071E E135       17 X13793:  DW       13793
0720 6400       18 X100:    DW       100
0722 8808       19 X2184:   DW       2184
0724 CC0C       20 X3276:   DW       3276
0726 0064       21 X25600:  DW       25600
0728 0010       22 X4096:   DW       4096
                23 ;
                24 ;
0000            25          ORG      0
                26 ;
0000 31FF3F     27        INITL: LXI SP,3FFFH
0003 21D301     28          LXI      H,SCALO ;SCALES RADIUS TO FEET
0006 CD9C02     29          CALL     FMATH   ;STORES AT RADIUS
0009 21E700     30          LXI      H,SQR
000C CD9C02     31          CALL     FMATH
000F 21F100     32          LXI      H,CALPI
0012 CD9C02     33          CALL     FMATH
0014            34          DA EQU 14H
0015 DB17       35          IN       DA+3      ;RESET D/A SYSTEM
0017 110000     36          LXI      D,0
001A 3E00       37          MVI      A,0
001C D314       38          OUT      DA
001E D315       39          OUT      DA+1
0020 7B         40          MOV      A,E
0021 D316       41          OUT      DA+2
0023 7A         42          MOV      A,D
0024 D317       43          OUT      DA+3
                44 ;
                45 ;
                46               ;PROGRAMMED A/D ST800
0010            47                 AD EQU 10H
0026 3E00       48 MAIN:           MVI A,0     ;CLEAR STATUS REGISTER
0028 D310       49                 OUT AD
002A 3E00       50                 MVI A,0     ;SET START CHANNEL NO=0
002C D311       51                 OUT AD+1
002E 3C         52                 INR A       ;SET END CHANNEL REG=1
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ      SEQ         SOURCE STATEMENT

002F D312      53              OUT AD+2
0031 D313      54              OUT AD+3      ;START CONVERSION
0033 DB10      55         LP1: IN AD         ;READ STATUS FOR CONVERSION
0035 0F        56              RRC           ;COMPLETE(LSB=1 AT COMPLETE)
0036 D23300    57              JNC LP1
0039 DB11      58              IN AD+1       ;GET LOW BYTE
003B 5F        59              MOV E,A       ;STORE IN E
003C DB11      60              IN AD+1       ;GET HIGH BYTE
003E E60F      61              ANI 0FH       ;ZEROS 4 HIGH BITS
0040 57        62              MOV D,A       ;STORE IN D
0041 D313      63              OUT AD+3      ;START SECOND CONVERSION
0043 DB10      64         LP2: IN AD         ;WAIT FOR COMPLETE
0045 0F        65              RRC
0046 D24300    66              JNC LP2
0049 DB11      67              IN AD+1       ;GET LOW BYTE
004B 6F        68              MOV L,A       ;STORE IN L
004C DB11      69              IN AD+1       ;GET HIGH BYTE
004E E60F      70              ANI 0FH       ;ZEROS 4 HIGH BITS
0050 67        71              MOV H,A       ;STORE IN H
               72          ;END OF A/D CONVERSION
0051 013303    73         LXI     B,819
0054 7C        74         MOV     A,H
0055 B8        75         CMP     B
0056 FA6100    76         JM      TOOSM
0059 C26400    77         JNZ     CPDE
005C 7D        78         MOV     A,L
005D B9        79         CMP     C
005E F26400    80         JP      CPDE
0061 213303    81 TOOSM:  LXI     H,819
0064 7A        82 CPDE:   MOV     A,D
0065 B8        83         CMP     B
0066 FA7100    84         JM      TOSM1
0069 C27400    85         JNZ     STVAL
006C 7B        86         MOV     A,E
006D B9        87         CMP     C
006E F27400    88         JP      STVAL
0071 113303    89 TOSM1:  LXI     D,819
0074 221F3C    90 STVAL:  SHLD DP
0077 EB        91              XCHG
0078 22223C    92              SHLD DEPTH
               93 ;
               94 ;
               95 ;
               96 ;START OF FLOW CALCULATION AND OUTPUT
               97 ;
               98 ;
007B 21E101    99          LXI     H,SCAL1 ;SCALES DEPTH FROM A/D
007E CD9C02   100          CALL    FMATH   ;TO 0 TO 1.5 FEET
0081 21F501   101          LXI     H,SCAL2 ;SCALES DP=0 TO 100 INCHES
0084 CD9C02   102          CALL    FMATH
              103 ;
0087 210301   104          LXI     H,POL0  ;CALCULATES (RAD-DEPTH)
008A CD9C02   105          CALL    FMATH   ;STORES AT IM1
008D 210D01   106          LXI     H,POL1  ;CALCULATES (RAD-DEPTH)/RAD
0090 CD9C02   107          CALL    FMATH   ;STORES AT IM2=IM1/RAD
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ      SEQ          SOURCE STATEMENT

              108 ;
              109 ;
              110 ;
0093 211701   111          LXI     H,POL2   ;CALCULATES FIXED POINT
0096 CD9C02   112          CALL    FMATH    ;IM3=DEPTH(2*RAD-DEPTH)
              113 ;
0099 2A313C   114          LHLD    IM3      ;TAKES SQUARE ROOT OF
009C CD0F02   115          CALL    SQRT     ;IM3, RESULT AT IM3
009F 22313C   116          SHLD    IM3
              117 ;
00A2 213101   118          LXI     H,POL3   ;CALC. (RAD-DEPTH)*IM3
00A5 CD9C02   119          CALL    FMATH    ;STORES AT IM5
00A8 214301   120          LXI     H,POL4   ;CALCULATES IM2 SQUARED
00AB CD9C02   121          CALL    FMATH    ;STORES AT IM2SQ
              122 ;
00AE 214D01   123          LXI     H,POL5   ;CALCULATES FOURTH PWR IM2
00B1 CD9C02   124          CALL    FMATH    ;STORES AT IM24T
              125 ;
00B4 215701   126          LXI     H,POL6   ;CALCULATES SIXTH PWR IM2
00B7 CD9C02   127          CALL    FMATH    ;STORES AT IM26T
              128 ;
00BA 216101   129          LXI     H,POL7   ;CALC. RAD SQRD*ARCSIN(RAD-DEPTH)/RAD)
00BD CD9C02   130          CALL    FMATH    ;STORES AT IM6
              131 ;
              132 ;
00C0 2A1F3C   133          LHLD    DP       ;CALC. SQR ROOT DP
00C3 CD0F02   134          CALL    SQRT     ;STORES SQR RT DP * 256 AT DP
00C6 221F3C   135          SHLD    DP
              136 ;
00C9 219F01   137          LXI     H,POL8   ;CALC. K*SQR ROOT DP
00CC CD9C02   138          CALL    FMATH    ;STORES AT IM7
              139 ;
00CF 21B901   140          LXI     H,POL9   ;CALC. FINAL FLOW SCALED
00D2 CD9C02   141          CALL    FMATH    ;TO 0 TO 4095 FOR OUTPUT
              142                           ;STORES AT FLOW
              143 ;
              144 ;
              145            ;PROGRAMMED D/A ROUTINE
00D5 3E00     146              MVI A,0      ;SET START CHANNEL=0
00D7 D314     147              OUT DA
00D9 D315     148              OUT DA+1
00DB 2A463C   149          LHLD    FLOW     ;GET FLOW IN H-L
00DE 7D       150              MOV A,L
00DF D316     151              OUT DA+2     ;OUTPUT LOW BYTE OF 1ST CHANNEL
00E1 7C       152              MOV A,H
00E2 D317     153              OUT DA+3     ;OUTPUT HIGH BYTE OF 1ST CHANNEL
00E4 C32600   154              JMP MAIN
              155 ;
              156 ;
              157 ;
              158 ;
              159 ;REVERSE POLISH STRINGS
              160 ;
              161 ;
00E7 1C3C     162 SQR:     DW      RADIUS
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

```
LOC  OBJ      SEQ           SOURCE STATEMENT

00E9 1C3C      163           DW      RADIUS
00EB 04FF      164           DW      RMULT
00ED 06FF      165           DW      REQFLT
00EF 253C      166           DW      RSQR
               167 ;
               168 ;
00F1 0007      169 CALPI:    DW      PI
00F3 00FF      170           DW      RFLOT
00F5 0A07      171           DW      X20000
00F7 00FF      172           DW      RFLOT
00F9 05FF      173           DW      RDIV
00FB 253C      174           DW      RSQR
00FD 04FF      175           DW      RMULT
00FF 06FF      176           DW      REQFLT
0101 283C      177           DW      PIRSQ
               178 ;
               179 ;
0103 1C3C      180 POLO:     DW      RADIUS
0105 223C      181           DW      DEPTH
0107 03FF      182           DW      RSUB
0109 06FF      183           DW      REQFLT
010B 2B3C      184           DW      IM1
               185 ;
               186 ;
010D 2B3C      187 POL1:     DW      IM1
010F 1C3C      188           DW      RADIUS
0111 05FF      189           DW      RDIV
0113 06FF      190           DW      REQFLT
0115 2E3C      191           DW      IM2
               192 ;
               193 ;
0117 0E07      194 POL2:     DW      X2
0119 00FF      195           DW      RFLOT
011B 1C3C      196           DW      RADIUS
011D 04FF      197           DW      RMULT
011F 223C      198           DW      DEPTH
0121 03FF      199           DW      RSUB
0123 223C      200           DW      DEPTH
0125 04FF      201           DW      RMULT
0127 1007      202           DW      X256
0129 00FF      203           DW      RFLOT
012B 04FF      204           DW      RMULT
012D 07FF      205           DW      REQFIX
012F 313C      206           DW      IM3
               207 ;
               208 ;
0131 313C      209 POL3:     DW      IM3
0133 00FF      210           DW      RFLOT
0135 2807      211           DW      X4096
0137 00FF      212           DW      RFLOT
0139 05FF      213           DW      RDIV
013B 2B3C      214           DW      IM1
013D 04FF      215           DW      RMULT
013F 06FF      216           DW      REQFLT
0141 343C      217           DW      IM5
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ          SOURCE STATEMENT

              218 ;
              219 ;
0143 2E3C     220 POL4:   DW      IM2
0145 2E3C     221         DW      IM2
0147 04FF     222         DW      RMULT
0149 06FF     223         DW      REQFLT
014B 373C     224         DW      IM2SQ
              225 ;
              226 ;
014D 373C     227 POL5:   DW      IM2SQ
014F 373C     228         DW      IM2SQ
0151 04FF     229         DW      RMULT
0153 06FF     230         DW      REQFLT
0155 3A3C     231         DW      IM24T
              232 ;
              233 ;
0157 3A3C     234 POL6:   DW      IM24T
0159 373C     235         DW      IM2SQ
015B 04FF     236         DW      RMULT
015D 06FF     237         DW      REQFLT
015F 3D3C     238         DW      IM26T
              239 ;
              240 ;
0161 1207     241 POL7:   DW      X15
0163 00FF     242         DW      RFLOT
0165 3D3C     243         DW      IM26T
0167 04FF     244         DW      RMULT
0169 1407     245         DW      X336
016B 00FF     246         DW      RFLOT
016D 05FF     247         DW      RDIV
016F 3A3C     248         DW      IM24T
0171 0C07     249         DW      X3
0173 00FF     250         DW      RFLOT
0175 04FF     251         DW      RMULT
0177 1607     252         DW      X40
0179 00FF     253         DW      RFLOT
017B 05FF     254         DW      RDIV
017D 02FF     255         DW      RADD
017F 373C     256         DW      IM2SQ
0181 1807     257         DW      X6
0183 00FF     258         DW      RFLOT
0185 05FF     259         DW      RDIV
0187 02FF     260         DW      RADD
0189 1A07     261         DW      X1
018B 00FF     262         DW      RFLOT
018D 02FF     263         DW      RADD
018F 2E3C     264         DW      IM2
0191 04FF     265         DW      RMULT
0193 253C     266         DW      RSQR
0195 04FF     267         DW      RMULT
0197 343C     268         DW      IM5
0199 02FF     269         DW      RADD
019B 06FF     270         DW      REQFLT
019D 403C     271         DW      IM6
              272 ;
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

```
LOC  OBJ     SEQ           SOURCE STATEMENT

              273 ;
019F 0407     274 POL8:    DW      K
01A1 00FF     275          DW      RFLOT
01A3 1C07     276          DW      X1000
01A5 00FF     277          DW      RFLOT
01A7 05FF     278          DW      RDIV
01A9 1F3C     279          DW      DP
01AB 00FF     280          DW      RFLOT
01AD 04FF     281          DW      RMULT
01AF 2807     282          DW      X4096
01B1 00FF     283          DW      RFLOT
01B3 05FF     284          DW      RDIV
01B5 06FF     285          DW      REQFLT
01B7 433C     286          DW      IM7
              287 ;
              288 ;
01B9 283C     289 POL9:    DW      PIRSQ
01BB 403C     290          DW      IM6
01BD 03FF     291          DW      RSUB
01BF 433C     292          DW      IM7
01C1 04FF     293          DW      RMULT
01C3 1E07     294          DW      X13793
01C5 00FF     295          DW      RFLOT
01C7 2007     296          DW      X100
01C9 00FF     297          DW      RFLOT
01CB 05FF     298          DW      RDIV
01CD 04FF     299          DW      RMULT
01CF 07FF     300          DW      REQFIX
01D1 463C     301          DW      FLOW
              302 ;
              303 ;
              304 ;
              305 ;
              306 ;
01D3 0207     307 SCAL0:   DW      RAD
01D5 00FF     308          DW      RFLOT
01D7 2007     309          DW      X100
01D9 00FF     310          DW      RFLOT
01DB 05FF     311          DW      RDIV
01DD 06FF     312          DW      REQFLT
01DF 1C3C     313          DW      RADIUS
              314 ;
              315 ;
01E1 223C     316 SCAL1:   DW      DEPTH
01E3 00FF     317          DW      RFLOT
01E5 0607     318          DW      ZERO
01E7 00FF     319          DW      RFLOT
01E9 03FF     320          DW      RSUB
01EB 2207     321          DW      X2184
01ED 00FF     322          DW      RFLOT
01EF 05FF     323          DW      RDIV
01F1 06FF     324          DW      REQFLT
01F3 223C     325          DW      DEPTH
              326 ;
              327 ;
```

```
LOC  OBJ     SEQ         SOURCE STATEMENT

01F5 1F3C    328 SCAL2:  DW      DP
01F7 00FF    329         DW      RFLOT
01F9 0607    330         DW      ZERO
01FB 00FF    331         DW      RFLOT
01FD 03FF    332         DW      RSUB
01FF 2407    333         DW      X3276
0201 00FF    334         DW      RFLOT
0203 05FF    335         DW      RDIV
0205 2607    336         DW      X25600
0207 00FF    337         DW      RFLOT
0209 04FF    338         DW      RMULT
020B 07FF    339         DW      REQFIX
020D 1F3C    340         DW      DP
             341 ;
             342 ;
             343 ;
             344 ;
             345         ;8080 SQUARE ROOT ROUTINE
             346         ;16 BIT ARGUMENT, 16 BIT RESULT
             347         ;ENTER WITH ARGUMENT IN HL
             348         ;RESULT RETURNED IN HL
020F 110000  349       SQRT: LXI D,0000H ;INITIALIZE EXTENDED ARGUMENT,
0212 01FFFF  350             LXI B,0FFFFH ;COMPLEMENTED RESULT, AND
0215 3E10    351             MVI A,10H    ;LOOP COUNT.
0217 F5      352             PUSH PSW     ;SAVE LOOP COUNT.
             353                          ;
0218 EB      354       LOOP: XCHG
0219 7C      355             MOV A,H
021A 65      356             MOV H,L
021B 6A      357             MOV L,D
021C 29      358             DAD H        ;DOUBLE LEFT SHIFT OF
021D 17      359             RAL          ;DOUBLE STRING DEHL
021E 29      360             DAD H        ;(CONTAINING CURRENT REMAINDER).
021F 17      361             RAL
0220 EB      362             XCHG
0221 29      363             DAD H
0222 29      364             DAD H
0223 5A      365             MOV E,D
0224 57      366             MOV D,A
             367                          ;
0225 F1      368             POP PSW      ;RETRIEVE LOOP COUNT.
0226 E5      369             PUSH H       ;SAVE LOW ORDER 16.
0227 60      370             MOV H,B      ;MOVE PARTIAL RESULT
0228 69      371             MOV L,C      ;TO HL.
0229 29      372             DAD H        ;SHIFT LEFT, SHIFTING IN A ONE.
022A 2C      373             INR L
022B 44      374             MOV B,H      ;SAVE SHIFTED PARTIAL RESULT.
022C 4D      375             MOV C,L
022D 29      376             DAD H        ;SHIFT COPY OF PARTIAL RESULT AGAIN.
022E 2C      377             INR L
022F 19      378             DAD D        ;SUBTRACT SHIFTED PARTIAL RESULT
             379                          ;FROM HIGH 16 OF CURRENT REMAINDER
             380                          ;(BY ADDING COMPLEMENT).
0230 D23502  381             JNC NOGO     ;TEST SUBTRACTION RESULT.
0233 0D      382             DCR C        ;TACK A ZERO ONTO COMPLEMENTED
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ                 SOURCE STATEMENT

                                             ;PARTIAL RESULT.
0234 EB                       XCHG           ;REPLACE HIGH ORDER 16 OF CURRENT
                                             ;REMAINDER WITH SUBTRACTION RESULT.
0235 3D                 NOGO: DCR A          ;DECREMENT AND TEST LOOP COUNT.
0236 CA3E02                   JZ DONE
0239 E1                       POP H
023A F5                       PUSH PSW       ;SAVE LOOP COUNT.
023B C31802                   JMP LOOP
023E E1                 DONE: POP H          ;REMOVE UNUSED DATA FROM STACK.
                                             ;
023F 78                       MOV A,B
0240 2F                       CMA
0241 67                       MOV H,A        ;COMPLEMENT RESULT AND STORE IN HL.
0242 79                       MOV A,C
0243 2F                       CMA
0244 6F                       MOV L,A
0245 C9                       RET
             ;
             ;
             ;
             ;
             ;
             ;REVERSE POLISH ARITHMETIC PROCESSOR
             ;CALLING FORMAT:      LXI H,POL1
             ;                     CALL FMATH
             ;WHERE POL1 IS THE START OF THE STRING TO BE PROCESSED
             ;NO STORAGE OF VARIABLES IS ALLOWED ABOVE FF00H
             ;ADDRESSES STARTING WITH FF INDICATE AN OPERATOR
             ;
             ;
0246 7B      OPRAT:  MOV     A,E
0247 FE06            CPI     6        ;CARRY SET IF E<6
0249 DA6902          JC      REALOP
024C 23      STRBK:  INX     H
024D 5E              MOV     E,M
024E 23              INX     H
024F 56              MOV     D,M      ;GET ADDRESS TO STR RESULT
0250 EB              XCHG
0251 D1              POP     D
0252 C1              POP     B
0253 48              MOV     C,B
0254 E5              PUSH    H
0255 FE06            CPI     6
0257 C26102          JNZ     FIXIT
025A E1              POP     H
025B 71              MOV     M,C
025C 23              INX     H
025D 72      XYZ:    MOV     M,D
025E 23              INX     H
025F 73              MOV     M,E
0260 C9              RET
0261 CD6003  FIXIT:  CALL    FIXX
0264 E1              POP     H
0265 73              MOV     M,E
0266 23              INX     H
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

```
LOC  OBJ     SEQ         SOURCE STATEMENT

0267 72                  MOV    M,D
0268 C9                  RET
             ;
0269 221A3C  REALOP:     SHLD   HLSTR
026C D1                  POP    D
026D C1                  POP    B
026E 48                  MOV    C,B      ;1ST IN CDE
026F 3D                  DCR    A
0270 FAB202              JM     FLOAT
0273 3D                  DCR    A
0274 FAB902              JM     UNARY
0277 32193C              STA    OPSTR    ;STORE OPERATOR-2
027A E1                  POP    H
027B F1                  POP    PSW
027C 47                  MOV    B,A      ;2ND IN BHL
027D E5                  PUSH   H        ;SAVE RETURN ADDRESS
027E 219502              LXI    H,ADNE   ;ON STACK
0281 E3                  XTHL
0282 3A193C              LDA    OPSTR    ;GET OPERATOR-2
0285 B7                  ORA    A        ;SET FLAGS
0286 CAC202              JZ     FADD
0289 3D                  DCR    A
028A CABF02              JZ     FSUB
028D 3D                  DCR    A
028E CA2604              JZ     FMPY
0291 3D                  DCR    A
0292 CAAE03              JZ     FDIV
0295 41      ADNE:       MOV    B,C
0296 C5                  PUSH   B        ;SAVE RESULT ON STACK
0297 D5                  PUSH   D        ;FOR NEXT OPERAND
0298 2A1A3C              LHLD   HLSTR
029B 23                  INX    H
029C 5E      FMATH:      MOV    E,M
029D 23                  INX    H
029E 7E                  MOV    A,M
029F FEFF                CPI    0FFH
02A1 CA4602              JZ     OPRAT
02A4 56                  MOV    D,M
02A5 1A                  LDAX   D
02A6 F5                  PUSH   PSW
02A7 13                  INX    D
02A8 1A                  LDAX   D
02A9 47                  MOV    B,A
02AA 13                  INX    D
02AB 1A                  LDAX   D
02AC 4F                  MOV    C,A
02AD C5                  PUSH   B
02AE 23                  INX    H
02AF C39C02              JMP    FMATH
             ;
02B2 59      FLOAT:      MOV    E,C
02B3 CD8C03              CALL   FLOT
02B6 C39502              JMP    ADNE
             ;
02B9 CDA403  UNARY:      CALL   FNEG
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ        SOURCE STATEMENT

02BC C39502  493        JMP     ADNE
             494 ;
             495 ;OPERANDS STORED ON STACK AS:
             496 ;      EXPONENT-????????
             497 ;        MANT1   -MANT2
             498 ;VALUE STORED IN MEMORY LOCATION AS:
             499 ;        LABEL: EXPONENT
             500 ;               MANT1
             501 ;               MANT2
             502 ;           OR FOR FIXED:
             503 ;          LABEL: LOW BYTE
             504 ;                 HIGH BYTE
             505 ;
             506 ;7-FIX RESULT AND STORE AT LAST ADDRESS
             507 ;6-STORE RESULT AT LAST ADDRESS
             508 ;5-DIVIDE
             509 ;4-MULTIPLY
             510 ;3-SUBTRACT
             511 ;2-ADD
             512 ;1-UNARY '-' (NEGATION)
             513 ;0-FLOAT
FF00         514                RFLOT EQU OFF00H
FF01         515                RNEG EQU OFF01H
FF02         516                RADD EQU OFF02H
FF03         517                RSUB EQU OFF03H
FF04         518                RMULT EQU OFF04H
FF05         519                RDIV EQU OFF05H
FF06         520                REQFLT EQU OFF06H
FF07         521                REQFIX EQU OFF07H
             522 ;
             523 ;FLOATING POINT ROUTINE FSUB: THIS ROUTINE SETS UP THE PARAMETERS TO PASS
             524 ;TO FADD WHICH RESULTS IN SUBTRACTION OF A NUMBER IN C-D-E FROM A
             525 ;NUMBER IN B-H-L.  IT MUST BE LOADED INTO MEMORY IMMEDIATELY PRECEEDING
             526 ;FADD.
02BF CDA403  527 FSUB:   CALL FNEG             ;NEGATE C-D-E
             528 ;FLOATING POINT ROUTINE FADD: THIS ROUTINE ADDS A FLOATING POINT
             529 ;NUMBER IN C-D-E TO A FLOATING POINT NUMBER IN B-H-L AND RETURNS WITH
             530 ;THE RESULT IN C-D-E. IT SHOULD BE LOADED INTO MEMORY IMMEDIATELY
             531 ;FOLLOWING FSUB.
02C2 CDBD05  532 FADD:   CALL ZRO              ;CHECK FOR ZERO C-D-E
02C5 87      533         ADD A
02C6 EB      534 FA00:   XCHG
02C7 79      535         MOV A,C        ;EXCHANGE C AND B
02C8 48      536         MOV C,B
02C9 47      537         MOV B,A
02CA C8      538         RZ                   ;ZERO IMPLIES ONE NUMBER WAS ZERO,
             539                               ;RESULT IN C-D-E
02CB CDBD05  540         CALL ZRO              ;CHECK NEXT VALUE (B-H-L)
02CE 87      541         ADD A
02CF CAC602  542         JZ FA00        ;ZERO IMPLIES B-H-L WAS ZERO,
             543                               ;RESULT = C-D-E
02D2 E5      544         PUSH H
02D3 21113C  545         LXI H,S2
02D6 79      546         MOV A,C
02D7 77      547         MOV M,A        ;RESERVE EXPONENT
```

```
LOC  OBJ     SEQ          SOURCE STATEMENT

02D8 E67F     548          ANI 7FH           ;MASK SIGN BIT
02DA 4F       549          MOV C,A
02DB 2B       550          DCX H
02DC 78       551          MOV A,B
02DD 77       552          MOV M,A
02DE E67F     553          ANI 7FH
02E0 47       554          MOV B,A
02E1 B9       555          CMP C              ;MATCH EXPONENTS IF NOT EQUAL
02E2 CA1103   556          JZ FA03           ;IF Z=1, EXPONENTS EQUAL, CONTINUE
02E5 DAF502   557          JC FA01           ;CARRY = 1 IMPLIES C>B
02E8 41       558          MOV B,C           ;C<B, SWITCH TO MAKE B<C
02E9 4F       559          MOV C,A
02EA C5       560          PUSH B
02EB 46       561          MOV B,M
02EC 23       562          INX H
02ED 4E       563          MOV C,M
02EE 70       564          MOV M,B
02EF 2B       565          DCX H
02F0 71       566          MOV M,C
02F1 C1       567          POP B
02F2 E1       568          POP H
02F3 EB       569          XCHG                  ;SWITCH MANTISSAE
02F4 E5       570          PUSH H
02F5 E1       571 FA01:    POP H
02F6 78       572          MOV A,B
02F7 3C       573 FA02:    INR A               ;INCREASE EXPONENT BY 1, NUMBER BY 2X
02F8 47       574          MOV B,A
02F9 37       575          STC                  ;ZERO CARRY
02FA 3F       576          CMC
02FB 7C       577          MOV A,H
02FC 1F       578          RAR                  ;DIVIDE HIGH MANTISSA BY 2
02FD 67       579          MOV H,A
02FE 7D       580          MOV A,L
02FF 1F       581          RAR                  ;DIVIDE LOW MANTISSA BY 2
0300 6F       582          MOV L,A
0301 AF       583          XRA A               ;CLEAR A TO SET UP FOR ROUNDOFF
0302 17       584          RAL                  ;DROPPED BIT FROM CARRY INTO LSB OF A
0303 32123C   585          STA S3
0306 78       586          MOV A,B
0307 B9       587          CMP C               ;EXPONENTS MATCHED?
0308 C2F702   588          JNZ FA02                ;IF Z=1, NO RETURN AND INCREMENT
030B 3A123C   589          LDA S3
030E 85       590          ADD L               ;ROUNDOFF
030F 6F       591          MOV L,A
0310 E5       592          PUSH H
0311 E1       593 FA03:    POP H               ;EXPONENTS NOW MATCHED, CHECK SIGN OF
              594                                   ;NUMBER
0312 3A113C   595          LDA S2
0315 E680     596          ANI 80H           ;MASK ALL BUT SIGN BIT
0317 4F       597          MOV C,A           ;STORE SIGN
0318 3A103C   598          LDA S1
031B 81       599          ADD C
031C F24C03   600          JP FA10           ;IF THE ANSWER IS +, SIGN SAME, CONTINUE
031F 7A       601 FA04:    MOV A,D           ;SIGH DIFFERENT, WHICH IS LARGER?
0320 BC       602          CMP H
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

```
LOC  OBJ     SEQ         SOURCE STATEMENT

0321 CA4103  603          JZ FA09           ;EQUAL, COMPARE E AND L
0324 DA2C03  604          JC FA06           ;C=1 IMPLIES H>D, COMPLIMENT C-D-E AND
             605                                    ;ITS SIGN
0327 EB      606 FA05:    XCHG                 ;D>H, SWITCH NUMBERS
0328 79      607          MOV A,C
0329 C32F03  608          JMP FA07
032C 79      609 FA06:    MOV A,C
032D C680    610          ADI 80H
032F 80      611 FA07:    ADD B                ;SET TOTAL EXPONENT
0330 4F      612          MOV C,A           ;EXPONENT IN C
0331 37      613          STC
0332 3F      614          CMC
0333 7B      615          MOV A,E           ;COMPLIMENT SMALLER (NOW IN CDE)
0334 2F      616          CMA
0335 5F      617          MOV E,A
0336 7A      618          MOV A,D
0337 2F      619          CMA
0338 57      620          MOV D,A
0339 13      621          INX D
033A 19      622          DAD D                 ;RESULT IN C-H-L, IGNORE OVERFLOW
033B EB      623          XCHG                   ;RESULT IN C-D-E
033C 0610    624 FA08:    MVI B,10H
033E C38E03  625          JMP FL1           ;GO TO LEFT JUSTIFY ROUTINE IN FLOT
             626                                    ;ROUTINE
0341 7B      627 FA09:    MOV A,E           ;D,H SAME, COMPARE E AND L
0342 BD      628          CMP L
0343 CA5A03  629          JZ FA11           ;EQUAL MAGNITUDE, OPPOSITE SIGN,
             630                                    ;RESULT ZERO
0346 DA2C03  631          JC FA06
0349 C32703  632          JMP FA05
034C 79      633 FA10:    MOV A,C           ;SIGN SAME, SET EXPONENT
034D 80      634          ADD B
034E 4F      635          MOV C,A
034F 19      636          DAD D                 ;ADD MANTISSAE
0350 EB      637          XCHG
0351 D0      638          RNC                    ;CARRY NOT SET, RETURN
0352 0C      639          INR C                 ;CARRY SET, INCREMENT EXPONENT
0353 7A      640          MOV A,D           ;LEFT JUSTIFY
0354 1F      641          RAR
0355 57      642          MOV D,A
0356 7B      643          MOV A,E
0357 1F      644          RAR
0358 5F      645          MOV E,A
0359 C9      646          RET
035A 0E40    647 FA11:    MVI C,40H                 ;ANSWER ZERO, RETURN
035C 110000  648          LXI D,0000H          ;SET ZERO ANSWER
035F C9      649          RET
             650 ;FLOATING POINT ROUTINE FIXX: THIS ROUTINE CONVERTS A FLOATING POINT
             651 ;NUMBER IN C-D-E TO A 16 BIT UNSIGNED FIXED POINT NUMBER IN D-E.
             652 ;FRACTIONS ARE TRUNCATED. UPON RETURN, THE STATE OF THE A REGISTER,
             653 ;WHEN DOUBLED, INDICATES THE CONDITION OF THE FIX.  A PROPER TEST
             654 ;ROUTINE WOULD CONSIST
             655 ;
             656 ;        CALL FIXX               ;CALL FIX ROUTINE
             657 ;        ADD A               ;DOUBLE A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ         SOURCE STATEMENT

             658 ;        JZ OK              ;IF RESULT ZERO, FIX PERFORMED OK
             659 ;        JM BN              ;IF NEGATIVE, IT TRIED TO FIX A NEGATIVE
             660                                   ;NUMBER
             661 ;        JP BP              ;IF PLUS, IT TRIED TO FIX A NUMBER > OR =
             662                                   ;2(16)
             663 ;                                 (CANNOT BE HANDLED IN 16 BITS)
0360 AF      664 FIXX:    XRA A
0361 67      665          MOV H,A
0362 6F      666          MOV L,A
0363 81      667          ADD C
0364 FA8103  668          JM FX2              ;IF MINUS, FAULT TO RESULT
0367 C640    669          ADI 40H
0369 F28403  670          JP FX3              ;IF +, THE NUMBER IS A FRACTION, RESULT
             671                                   ;ZERO
036C E63F    672          ANI 3FH            ;MASK THE 2 MSBS OF THE EXPONENT
036E FE11    673          CPI 11H
0370 D28903  674          JNC FX4            ;OVERLOADED IF >16
0373 4F      675          MOV C,A
0374 CDAD05  676 FX1:     CALL DROT                ;ROTATE D-E
0377 CDB605  677          CALL HROT                ;ROTATE H-L FROM D-E
037A 0D      678          DCR C
037B C27403  679          JNZ FX1
037E AF      680          XRA A               ;CLEAR A IMPLIES RESULT SOUND
037F EB      681          XCHG                 ;RESULT IN D-E
0380 C9      682          RET
0381 3E40    683 FX2:     MVI A,40H                ;WHEN DOUBLED, RESULT MINUS, IMPLIES
             684                                   ;MINUS FIX ATT
0383 C9      685          RET
0384 110000  686 FX3:     LXI D,0000H           ;RESULT ZERO
0387 7A      687          MOV A,D
0388 C9      688          RET
0389 3E01    689 FX4:     MVI A,01H                ;OVERFLOW, + A WHEN DOUBLED
038B C9      690          RET
             691 ;FLOATING POINT ROUTINE FLOT: THIS ROUTINE CONVERTS AN UNSIGNED, 16 BIT
             692 ;NUMBER IN D-E TO A FLOATING POINT NUMBER AND RETURNS WITH THE RESULT
             693 ;OF THE FLOAT IN C(EXP)-D-E.  THE UNSIGNED NUMBER, UPON FLOATING, IS
             694 ;ASSIGNED A POSITIVE VALUE.  IF A NEGATIVE VALUE IS DESIRED, THE
             695 ;NEGATE ROUTINE FNEG SHOULD BE CALLED UPON RETTRN.
038C 0E50    696 FLOT:    MVI C,50H                ;C=EXPONENT OF +16
038E CDC105  697 FL1:     CALL ZRDE                ;SEE IF D=E=0
0391 87      698          ADD A
0392 CAA103  699          JZ FL3
0395 AF      700 FL2:     XRA A
0396 82      701          ADD D
0397 3E00    702          MVI A,00H                ;ZERO A TO INDICATE A PROPER RETURN
0399 F8      703          RM
039A CDAD05  704          CALL DROT                ;ROTATE D-E
039D 0D      705          DCR C               ;DECREASE EXPONENT BY 1
039E C39503  706          JMP FL2            ;SEE IF LEFT JUSTIFIED YET
03A1 0E40    707 FL3:     MVI C,40H                ;RESULT ZERO, SET PROPER EXPONENT
03A3 C9      708          RET
             709 ;FLOATING POINT ROUTINE FNEG: THIS ROUTINE NEGATES A FLOATING POINT
             710 ;NUMBER WHOSE EXPONENT IS IN C. IF C-D-E HAS A VALUE OF ZERO,
             711 ;NO OPERATION IS PERFORMED.
03A4 CDBD05  712 FNEG:    CALL ZRO                 ;CHECK FOR ZERO VALUE
```

```
LOC  OBJ     SEQ         SOURCE STATEMENT

03A7 87       713          ADD A
03A8 C8       714          RZ                     ;IF C-D-E IS ZERO, NO NEGATION PERFORMED
03A9 3E80     715 FN1:     MVI A,80H
03AB 81       716          ADD C               ;COMPLIMENT SIGN BIT
03AC 4F       717          MOV C,A
03AD C9       718          RET
              719 ;FLOATING POINT ROUTINE FDIV: THIS ROUTINE DIVIDES A FLOATING POINT
              720 ;NUMBER IN B-H-L BY A FLOATING POINT NUMBER IN C-D-E BY THE NONRESTORING
              721 ;METHOD AND RETURNS WITH THE RESULT IN C-D-E. IF A DIVISION BY ZERO IS
              722 ;ATTEMPTED, THE ROUTINE RETURNS WITH A '1' IN THE A REGISTER.
              723 ;OTHERWISE, THE RETURN IS MADE WITH A ZERO A REGISTER.
03AE CDBD05   724 FDIV:    CALL ZRO                ;CHECK FOR DIVISION BY ZERO
03B1 87       725          ADD A
03B2 3E01     726          MVI A,01H
03B4 C8       727          RZ
03B5 E5       728 FD00:    PUSH H
03B6 21133C   729          LXI H,S4
03B9 AF       730          XRA A               ;ZERO S4 AND S5
03BA 77       731          MOV M,A
03BB 23       732          INX H
03BC 77       733          MOV M,A
03BD 23       734          INX H
03BE 3611     735          MVI M,11H               ;17 IN ME
03C0 23       736          INX H
03C1 3601     737          MVI M,01H               ;FIRST POINTER IN MH
03C3 23       738          INX H
03C4 78       739          MOV A,B
03C5 91       740          SUB C               ;DIVIDE EXPONENTS
03C6 C641     741          ADI 41H            ;SET OFFSET
03C8 E67F     742          ANI 7FH            ;MASK SIGN BIT
03CA 77       743          MOV M,A            ;STORE IN ML
03CB 79       744          MOV A,C
03CC E680     745          ANI 80H            ;MASK THE REST
03CE 4F       746          MOV C,A
03CF 78       747          MOV A,B
03D0 E680     748          ANI 80H            ;SET UP SIGN
03D2 81       749          ADD C
03D3 86       750          ADD M
03D4 77       751          MOV M,A
03D5 AF       752          XRA A
03D6 47       753          MOV B,A
03D7 4F       754          MOV C,A
03D8 E1       755          POP H
03D9 B9       756 FD01:    CMP C
03DA 37       757          STC
03DB C2DF03   758          JNZ FD02                ;NOT ZERO IMPLIES SIGNS DIFFERENT,
              759                                  ;Q(N)=0
03DE 3F       760          CMC
03DF 3F       761 FD02:    CMC                   ;CARRY CONTAINS THE CURRENT QUOTIENT BIT
03E0 E5       762          PUSH H
03E1 21133C   763          LXI H,S4
03E4 7E       764          MOV A,M
03E5 17       765          RAL
03E6 77       766          MOV M,A
03E7 23       767          INX H
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ        SOURCE STATEMENT

03E8 7E       768            MOV A,M
03E9 17       769            RAL
03EA 77       770            MOV M,A
03EB 23       771            INX H
03EC 35       772            DCR M
03ED CA1604   773            JZ FD06          ;WHEN ZERO, DONE, SIGN BIT IN CARRY
03F0 23       774            INX H
03F1 35       775            DCR M
03F2 E1       776            POP H
03F3 CAF903   777            JZ FD03          ;ZERO IMPLIES FIRST PASS
03F6 29       778            DAD H              ;DOUBLE DIVIDENT OR REMAINDER
03F7 17       779            RAL
03F8 47       780            MOV B,A
03F9 3A133C   781 FD03:      LDA S4
03FC E601     782            ANI 01H          ;MASK ALL BUT MORE RECENT Q BIT
03FE 87       783            ADD A
03FF CA0D04   784            JZ FD04          ;Q(N)=0 IMPLIES GO ADD 2R AND Y
0402 7D       785            MOV A,L          ;Q(N)=1 IMPLIES SUBTRACT Y FROM 2R
0403 93       786            SUB E
0404 6F       787            MOV L,A
0405 7C       788            MOV A,H
0406 9A       789            SBB D
0407 67       790            MOV H,A
0408 78       791            MOV A,B
0409 99       792            SBB C
040A C31004   793            JMP FD05
040D 19       794 FD04:      DAD D              ;RESULT IN H-L
040E 78       795            MOV A,B
040F 89       796            ADC C
0410 E601     797 FD05:      ANI 01H          ;MASK ALL BUT SIGN
0412 47       798            MOV B,A
0413 C3D903   799            JMP FD01
0416 E1       800 FD06:      POP H              ;CLEAR THE STACK
0417 21173C   801            LXI H,ML
041A 4E       802            MOV C,M
041B 2B       803            DCX H
041C 2B       804            DCX H
041D 2B       805            DCX H
041E 56       806            MOV D,M   ;QUOTIENT LSB IN E
041F 2B       807            DCX H
0420 5E       808            MOV E,M          ;QUOTIENT MSB IND
0421 0610     809            MVI B,10H               ;BEGIN TO LEFT JUSTIFY
0423 C38E03   810            JMP FL1          ;LEFT JUSTIFY IN FLOT ROUTINE
              811 ;FLOATING POINT ROUTINE FMPY: THIS ROUTINE MULTIPLIES A FLOATING
              812 ;POINT NUMBER IN B-H-L BY A FLOATING POINT NUMBER IN C-D-E
              813 ;AND RETURNS WITH THE RESULT IN C-D-E. IT DOES NOT CHECK FOR
              814 ;EXPONENTIAL OVERFLOW.
0426 78       815 FMPY:      MOV A,B          ;STORE ONE MULTIPLIER
0427 32153C   816            STA ME
042A 7C       817            MOV A,H
042B 32163C   818            STA MH
042E 7D       819            MOV A,L
042F 32173C   820            STA ML
0432 21103C   821            LXI H,S1                ;ADDRESS OF S1
0435 AF       822            XRA A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ       SEQ        SOURCE STATEMENT

0436 77        823         MOV M,A
0437 77        824         MOV M,A           ;ZERO S1-S5
0438 23        825         INX H
0439 77        826         MOV M,A           ;S1-S4 TEMPORARY SUM STORAGE
043A 23        827         INX H
043B 77        828         MOV M,A
043C 23        829         INX H
043D 77        830         MOV M,A
043E 23        831         INX H
043F 77        832         MOV M,A           ;S5 IS FINISH POINTER
0440 23        833         INX H
0441 7E        834         MOV A,M           ;BRING IN EXPONENT OF MULTIPLIER
0442 E680      835         ANI 80H           ;MASK ALL BUT SIGN BIT
0444 47        836         MOV B,A
0445 7E        837         MOV A,M
0446 81        838         ADD C               ;MULTIPLY EXPONENTS, SET SIGN
0447 C640      839         ADI 40H          ;SET OFFSET
0449 E67F      840         ANI 7FH           ;MASK SIGN
044B 77        841         MOV M,A
044C 79        842         MOV A,C
044D E680      843         ANI 80H           ;MASK ALL BUT SIGN BIT
044F 80        844         ADD B               ;SET SIGN IN EXPONENT
0450 86        845         ADD M
0451 77        846         MOV M,A           ;RESULT IN ME
0452 23        847         INX H
0453 23        848         INX H
0454 46        849         MOV B,M           ;LSB MULTIPLIER IN B
0455 0E08      850         MVI C,08H                ;DECREMENTER = 8
0457 210000    851         LXI H,0000H            ;ZERO H-L
045A 37        852         STC
045B 3F        853         CMC
045C C36504    854         JMP FM2
045F CDAD05    855 FM1:    CALL DROT                ;ROTATE D-E PAIR
0462 CDB605    856         CALL HROT                ;ROTATE H-L FROM D-E
0465 78        857 FM2:    MOV A,B           ;ADDEND NOW IN H-L-D-E
0466 1F        858         RAR                   ;CURRENT MULTIPLIER IN CARRY
0467 47        859         MOV B,A
0468 D27F04    860         JNC FM3           ;BIT ZERO, DONT ADD
046B C5        861         PUSH B             ;STORE B-C
046C 01103C    862         LXI B,S1                 ;S1 ADDRESS IN B-C
046F 0A        863         LDAX B             ;SUM
0470 83        864         ADD E
0471 02        865         STAX B
0472 03        866         INX B
0473 0A        867         LDAX B
0474 8A        868         ADC D
0475 02        869         STAX B
0476 03        870         INX B
0477 0A        871         LDAX B
0478 8D        872         ADC L
0479 02        873         STAX B
047A 03        874         INX B
047B 0A        875         LDAX B
047C 8C        876         ADC H
047D 02        877         STAX B             ;ALL SUMMED IN (NSB)S4-S3-S2-S1
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ      SEQ        SOURCE STATEMENT

047E C1        878          POP B             ;RESTORE B-C
047F 0D        879 FM3:     DCR C
0480 C25F04    880          JNZ FM1          ;FINISH MULTIPLIER
0483 01143C    881          LXI B,S5                ;S5 ADDRESS IN B-C
0486 0A        882          LDAX B
0487 3D        883          DCR A
0488 CA9704    884          JZ FM4            ;Z=1 IMPLIES ALL DONE, LEFT JUSTIFY
048B 3C        885          INR A
048C 3C        886          INR A
048D 02        887          STAX B            ;SET S5 TO INDICATE LAST PASS COMPLETE
048E 03        888          INX B
048F 03        889          INX B
0490 0A        890          LDAX B           ;MH TO A
0491 47        891          MOV B,A          ;MH TO B, MULTIPLY BY HIGH WORD
0492 0E08      892          MVI C,08H               ;RESTORE POINTER TO C
0494 C35F04    893          JMP FM1          ;FINISH MULTIPLICATION, MSB MULTIPLIER
0497 21153C    894 FM4:     LXI H,ME                ;SUM NOW IN S4-S1
049A 7E        895          MOV A,M
049B 2B        896          DCX H
049C 2B        897          DCX H             ;S4 ADDRESS
049D 46        898          MOV B,M          ;S4 IN B
049E 2B        899          DCX H              ;S3 IN C
049F 4E        900          MOV C,M
04A0 2B        901          DCX H
04A1 56        902          MOV D,M          ;S2 IN D
04A2 2B        903          DCX H
04A3 5E        904          MOV E,M          ;S1 IN E
04A4 60        905          MOV H,B
04A5 69        906          MOV L,C
04A6 4F        907          MOV C,A          ;EXPONENT IN C
04A7 0610      908          MVI B,10H
04A9 AF        909 FM5:     XRA A
04AA 84        910          ADD H
04AB FABC04    911          JM FM6            ;IF MINUS, LEFT JUSTIFIED, CONTINUE
04AE CDAD05    912          CALL DROT               ;ROTATE D-E
04B1 CDB605    913          CALL HROT               ;ROTATE H-L
04B4 0D        914          DCR C              ;REDUCE EXPONENT
04B5 05        915          DCR B
04B6 C2A904    916          JNZ FM5
04B9 C35A03    917          JMP FA11                ;RESULT ZERO, RETURN BY WAY OF ZERO
               918                                  ;ANSWER
04BC EB        919 FM6:     XCHG                ;RESULT IN C-D-E
04BD C9        920          RET
               921 ;FLOATING POINT ROUTINE FBCD: THIS ROUTINE CONVERTS A
               922 ;FLOATING POINT NUMBER IN C-D-E TO A FLOATING POINT
               923 ;BCD NUMBER IN C-D-E.  IT RETURNS WIT THE REGISTERS ASSIGNED AS FOLLOWS:
               924 ;
               925 ;        REGISTER                      BIT
               926 ;                                7  6  5  4  3  2  1  0
               927 ;          C                  SN SE EH EH EL EL EL EL
               928 ;          D                  X4 X4 X4 X4 X3 X3 X3 X3
               929 ;          E                  X2 X2 X2 X2 X1 X1 X1 X1
               930 ;
               931 ;WHERE SN IS THE SIGN OF THE NUMBER (0 IMPLIES PLUS), SE IS
               932 ;THE SIGN OF THE EXPONENT, EH IS THE TWO BIT BCD UPPER EXPONENT DIGIT,
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

```
LOC  OBJ     SEQ         SOURCE STATEMENT

              933 ;EL IS THE BCD LOWER EXPONENT DIGIT, AND X4 AND X1 IS THE BCD MANTISSA
              934 ;AS X.XXX.
              935 ;
04BE 79       936 FBCD:    MOV A,C
04BF E680     937          ANI 80H               ;MASK ALL BUT SIGN
04C1 F5       938          PUSH PSW
04C2 3E03     939          MVI A,03H                    ;3 IMPLIES EXPONENT OF 10(3)
04C4 32183C   940          STA MX
04C7 79       941          MOV A,C
04C8 E67F     942          ANI 7FH               ;MASK SIGN
04CA 4F       943          MOV C,A
04CB 3E4E     944 FB1:     MVI A,4EH
04CD B9       945          CMP C
04CE DAE304   946          JC FB2                 ;C IMPLIES 4E<REG C, DIVIDE BY 10
04D1 C2F504   947          JNZ FB3               ;NZ IMPLIES THE NUMBER MAY BE TOO
              948                                       ;SMALL, CHECK
04D4 3E9C     949          MVI A,9CH
04D6 BA       950          CMP D
04D7 DAE304   951          JC FB2                 ;C IMPLIES 9C<REG D, DIVIDE BY 10
04DA C21705   952          JNZ FB4               ;NZ IMPLIES D<9C, NUMBER OK
04DD 3E3C     953          MVI A,3CH
04DF BB       954          CMP E
04E0 D21705   955          JNC FB4               ;NC IMPLIES 3C> OR = REG E, NUMBER OK
04E3 063D     956 FB2:     MVI B,3DH                    ;LOAD 0.1
04E5 26CC     957          MVI H,0CCH
04E7 6C       958          MOV L,H
04E8 CD2604   959          CALL FMPY                    ;X 0.1
04EB 3A183C   960          LDA MX
04EE 3C       961          INR A
04EF 32183C   962          STA MX                 ;INCREMENT EXPONENT POINTER
04F2 C3CB04   963          JMP FB1
04F5 3E4A     964 FB3:     MVI A,4AH
04F7 B9       965          CMP C
04F8 DA1705   966          JC FB4                 ;C IMPLIES 4A<REG C, NUMBER IS OK
04FB C20405   967          JNZ FB3A                     ;N2 IMPLIES REG C < 4AH, MULT
04FE 3EF9     968          MVI A,0F9H
0500 BA       969          CMP D
0501 DA1705   970          JC FB4                 ;C IMPLIES F9<REG D, NUMBER IS OK
0504 0644     971 FB3A:    MVI B,44H                    ;LOAD 10
0506 26A0     972          MVI H,0A0H
0508 2E00     973          MVI L,00H
050A CD2604   974          CALL FMPY                    ;X 10
050D 3A183C   975          LDA MX
0510 3D       976          DCR A
0511 32183C   977          STA MX                 ;DECREMENT EXPONENT
0514 C3F504   978          JMP FB3
0517 CD3C05   979 FB4:     CALL TROT                    ;NOTATE D-E TO H-L
051A AF       980          XRA A
051B CD4D05   981          CALL BCD                     ;CONVERT H-L TO BCD, RETURN WITH BCD
              982                                       ;IN H-L
051E F1       983          POP PSW               ;RESTORE # SIGN TO A
051F E5       984          PUSH H                 ;SAVE BCD
0520 4F       985          MOV C,A               ;PUT SIGN OF NUMBER IN C
0521 3A183C   986          LDA MX
0524 47       987          MOV B,A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ        SOURCE STATEMENT

0525 AF       988         XRA A
0526 80       989         ADD B
0527 F23405   990         JP FB5          ;P IMPLIES EXPONENT IS +
052A 2F       991         CMA
052B 3C       992         INR A            ;CHANGE MINUS EXPONENT TO 2S
              993                               ;COMPLIMENT
052C 67       994         MOV H,A
052D 3E40     995         MVI A,40H
052F 81       996         ADD C
0530 4F       997         MOV C,A         ;SET SIGN OF EXPONENT = -
0531 C33505   998         JMP FB6
0534 67       999 FB5:    MOV H,A
0535 3E07    1000 FB6:    MVI A,07H             ;SIGN OF # AND EXPONENT IN C, EXPONENT
             1001                               ;IN H
0537 CD4D05  1002         CALL BCD              ;RETURN WITH EXPONENT BCD AND ALL
             1003                               ;SIGNS IN C
053A D1      1004         POP D            ;RESTORE BCD MANTISSA TO D-E
053B C9      1005         RET
             1006 ;SUBROUTINE TROT:THIS SUBROUTINE ROTATES D-E AND H-L UNTIL THE
             1007 ;NUMERICAL EXPONENT IS REDUCED TO ZERO.
053C 210000  1008 TROT:   LXI H,0000H
053F CDAD05  1009 TRT1:   CALL DROT
0542 CDB605  1010         CALL HROT
0545 0D      1011         DCR C
0546 3E40    1012         MVI A,40H
0548 B9      1013         CMP C
0549 C8      1014         RZ
054A C33F05  1015         JMP TRT1
             1016 ;SUBROUTINE BCD: THIS SUBROUTINE CONVERTS A BINARY NUMBER IN
             1017 ;H-L TO A BCD NUMBER IN H-L. IF IT IS ENTERED WITH THE A REGISTER
             1018 ;EQUAL TO ZERO, IT CONVERTS THE NUMBERICAL MANTISSA IN H-L. IF THE
             1019 ;A REGISTER IS NONZERO, IT CONVERTS THE NUMERICAL EXPONENT IN H.
054D F5      1020 BCD:    PUSH PSW
054E 110000  1021         LXI D,0000H
0551 EB      1022         XCHG
0552 87      1023         ADD A
0553 CA5F05  1024         JZ BC1          ;Z IMPLIES MANTISSA TO BE CONVERTED
0556 0606    1025         MVI B,06H
0558 7A      1026         MOV A,D
0559 17      1027         RAL
055A 17      1028         RAL
055B 57      1029         MOV D,A
055C C36105  1030         JMP BC2
055F 0610    1031 BC1:    MVI B,10H
0561 CDAD05  1032 BC2:    CALL DROT
0564 CDB605  1033         CALL HROT
0567 05      1034         DCR B
0568 CAA605  1035         JZ BC6
056B 7D      1036         MOV A,L         ;CHECK FIRST DECADE
056C E60F    1037         ANI 0FH
056E FE05    1038         CPI 05H
0570 DA7B05  1039         JC BC3          ;C IMPLIES DECADE <5, SKIP
0573 3E03    1040         MVI A,03H
0575 85      1041         ADD L
0576 6F      1042         MOV L,A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ     SEQ          SOURCE STATEMENT

0577 3E00    1043          MVI A,00H
0579 8C      1044          ADC H
057A 67      1045          MOV H,A
057B 7D      1046 BC3:     MOV A,L           ;CHECK SECOND DECADE
057C E6F0    1047          ANI      OFOH
057E FE50    1048          CPI 50H
0580 DA8B05  1049          JC BC4             ;C IMPLIES DECADE <5, SKIP
0583 3E30    1050          MVI A,30H
0585 85      1051          ADD L
0586 6F      1052          MOV L,A
0587 3E00    1053          MVI A,00H
0589 8C      1054          ADC H
058A 67      1055          MOV H,A
058B 7C      1056 BC4:     MOV A,H           ;CHECK THIRD DECADE
058C E60F    1057          ANI OFH
058E FE05    1058          CPI 05H
0590 DA9705  1059          JC BC5             ;C IMPLIES DECADE <5, SKIP
0593 3E03    1060          MVI A,03H
0595 84      1061          ADD H
0596 67      1062          MOV H,A
0597 7C      1063 BC5:     MOV A,H           ;CHECK FOURTH DECADE
0598 E6F0    1064          ANI OFOH
059A FE50    1065          CPI 50H
059C DA6105  1066          JC BC2             ;C IMPLIES DECADE <5, SKIP
059F 3E30    1067          MVI A,30H
05A1 84      1068          ADD H
05A2 67      1069          MOV H,A
05A3 C36105  1070          JMP BC2
05A6 F1      1071 BC6:     POP PSW
05A7 87      1072          ADD A
05A8 C8      1073          RZ                      ;Z IMPLIES MANTISSA, RETURN WITH BCD
             1074                                    ;IN H-L
05A9 79      1075          MOV A,C
05AA 85      1076          ADD L
05AB 4F      1077          MOV C,A
05AC C9      1078          RET
             1079 ;SUBROUTINE DROT: THIS SUBROUTINE ROTATES THE D-E REGISTER PAIR LEFT
05AD 37      1080 DROT:    STC
05AE 3F      1081          CMC
05AF 7B      1082          MOV A,E
05B0 17      1083          RAL
05B1 5F      1084          MOV E,A
05B2 7A      1085          MOV A,D
05B3 17      1086          RAL
05B4 57      1087          MOV D,A
05B5 C9      1088          RET
             1089 ;SUBROUTINE HROT: THIS SUBROUTINE ROTATES THE H-L REGISTER PAIR
             1090 ;LEFT, PUTTING THE CARRY INTO THE LSB OF L
05B6 7D      1091 HROT:    MOV A,L
05B7 17      1092          RAL
05B8 6F      1093          MOV L,A
05B9 7C      1094          MOV A,H
05BA 17      1095          RAL
05BB 67      1096          MOV H,A
05BC C9      1097          RET
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ    SEQ         SOURCE STATEMENT

            1098 ;SUBROUTINES ZRO AND ZRDE: THESE SUBROUTINES CHECK FOR A
            1099 ;ZERO VALUE FOR A FLOATING POINT NUMBER IN C-D-E (ZRO) OR
            1100 ;ZERO D-E ONLY (ZRDE).  IF THE RESULT WAS ZERO, THE RETURN
            1101 ;IS MADE WITH A ZERO A REGISTER.  OTHERWISE, A=40H UPON RETURN.
05BD 3E40   1102 ZRO:     MVI A,40H
05BF B9     1103          CMP C            ;CHECK FOR EXPONENT CORRESPONDING TO 0
05C0 C0     1104          RNZ                ;NZ IMPLIES NUMBER IS NOT ZERO
05C1 7B     1105 ZRDE:    MOV A,E        ;TEST D-E
05C2 82     1106          ADD D
05C3 DACB05 1107          JC ZR1          ;C = 1 IMPLIES NUMBER IS NOT 0
05C6 C2CB05 1108          JNZ ZR1        ;NZ IMPLIES NUMBER IS NOT ZERO
05C9 AF     1109          XRA A
05CA C9     1110          RET                ;RESULT WAS ZERO, RETURN WITH ZERO A
05CB 3E40   1111 ZR1:     MVI A,40H
05CD C9     1112          RET                ;RESULT NOT ZERO, RETURN WITH A=40H
            1113 ;
            1114 ;
            1115 ;
            1116 ;
            1117 ;
            1118 ;
            1119 ;
            1120 ;
            1121 ;RAM AREA HERE
            1122 ;
            1123 ;
3C10        1124          ORG     3C10H
            1125 ;
3C10 00     1126 S1:      DB      0
3C11 00     1127 S2:      DB      0
3C12 00     1128 S3:      DB      0
3C13 00     1129 S4:      DB      0
3C14 00     1130 S5:      DB      0
3C15 00     1131 ME:      DB      0
3C16 00     1132 MH:      DB      0
3C17 00     1133 ML:      DB      0
3C18 00     1134 MX:      DB      0
            1135          ;
            1136 ;
3C19 00     1137 OPSTR:   DB      0
3C1A 0000   1138 HLSTR:   DW      0
3C1C 00     1139 RADIUS:  DB      0,0,0
3C1D 00
3C1E 00
3C1F 00     1140 DP:      DB      0,0,0
3C20 00
3C21 00
3C22 00     1141 DEPTH:   DB      0,0,0
3C23 00
3C24 00
3C25 00     1142 RSQR:    DB      0,0,0
3C26 00
3C27 00
3C28 00     1143 PIRSQ:   DB      0,0,0
3C29 00
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0

LOC  OBJ    SEQ         SOURCE STATEMENT

3C2A 00
3C2B 00     1144 IM1:     DB      0,0,0
3C2C 00
3C2D 00
3C2E 00     1145 IM2:     DB      0,0,0
3C2F 00
3C30 00
3C31 00     1146 IM3:     DB      0,0,0
3C32 00
3C33 00
3C34 00     1147 IM5:     DB      0,0,0
3C35 00
3C36 00
3C37 00     1148 IM2SQ:   DB      0,0,0
3C38 00
3C39 00
3C3A 00     1149 IM24T:   DB      0,0,0
3C3B 00
3C3C 00
3C3D 00     1150 IM26T:   DB      0,0,0
3C3E 00
3C3F 00
3C40 00     1151 IM6:     DB      0,0,0
3C41 00
3C42 00
3C43 00     1152 IM7:     DB      0,0,0
3C44 00
3C45 00
3C46 00     1153 FLOW:    DB      0,0,0
3C47 00
3C48 00
            1154 ;
            1155 ;
            1156 ;
            1157          END
```

VITA

Leland Wayne Barker was born on October 29, 1939, in Kingsport, Tennessee. After graduating from Dobyns-Bennett High School in Kingsport in 1957, he attended East Tennessee State University until December 1958, while majoring in chemistry. From 1959 to 1960 he worked at Bausch and Lomb Optical Company. From 1960 to 1966 he was employed by Tennessee Eastman Company as a pipefitter. He enrolled at The University of Tennessee, Knoxville, in June 1966 and received a B.S. degree in Engineering Physics in June 1968. While at The University of Tennessee, he worked in the Stress Analysis Laboratory in the Mechanical Engineering Department. In the fall of 1968 he accepted a National Defense Education Act Fellowship at The University of Florida and began study toward a Master's degree in Physics. This degree was awarded in December 1970.

He has been employed in the chemical industries as a Physicist or an Electrical Engineer since January 1971. His job duties have included process instrumentation and process analyzers applications and development. In September 1976 he became a licensed professional engineer. He entered The University of Tennessee Graduate School, Knoxville, in September 1977.

He has been married to the former Chris McConnell since 1961 and is the father of two daughters, Cathy and Cindy.