MICROCOMPUTER BASED HEAT CONTROLLER

by

MAZEN AKKAM

B. S., Kansas State University, 1983

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

Approved by:

S S Hoft

Major Professor

TABLE OF CONTENTS

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

Page

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The Microcomputer Based Heat Controller, MBHC, is an interface circuit used to control temperatures in chemical reactors. It was designed and built at the Chemical Engineering Department at Kansas State University under the supervision of the Department Head, Dr. L. T. Fan and Professor Walter P. Walawender. The goal of this project was to monitor and control the reactor's temperatures with a computer. The computer monitors temperatures inside the reactor and also supplies appropriate messages to the user. MBHC is capable of measuring up to sixteen temperatures. The number can be increased if needed.

The purpose of this paper is to explain the design and operation of the control circuit. The reader should be able to get a complete understanding of how this circuit works, and then be able to operate and maintain it. The heart of MBHC is a Commodore 64 microcomputer which replaces analog controllers of the reactors. The microcomputer is interfaced to the reactor with integrated circuits. A 12-bit Analog to Digital Convertor, Sample and Hold, Analog switches, CMOS Latches, TTL, and LSTTL gates, and Peripheral Interface Adaptors are used.

This paper consists of five major sections, namely,

1) project development
2) hardware description
3) software description
4) future expansion
5) procedure of operation

PROJECT DEVELOPMENT

Before exploring the hardware or software of the system, some light will be directed upon the design process and the steps that lead to the final form of the project. A description of the system will give a better understanding of the problem.

A. Description of the Control Problem

1. Chemical process. The idea was to build a digital control system for a "Chemical reactor". Little concern was given to the type of chemical process going on inside the reactor. But it was known, from the beginning, that everything is being done to achieve certain chemical reactions (such as burning coal to produce natural gas) that require a specified temperature. So the main goal was to keep the temperature at a desired level with minimum deviation.

Note: For more information about the chemical process refer to Appendix A and references.

2. Description of the reactors. There are two reactors of different sizes. Most of the work was being done on the small reactor but the system can be expanded easily to control the other. A full description of the small reactor is given in Appendix A.

B. Nature of the system

The system involved is a complicated thermodynamic process that depends on many variables and factors such as mass change, momentum, and kinetic energy of the materials inside the reactor. It also depends on the external disturbances coming from the surroundings and the heaters. A very sophisticated mathematical model is needed to take all these factors into account. Some assumptions had to be made to simplify the problem. These assumptions are:

a. The disturbances from the surroundings in the form of heat exchange are neglected.

b. The system temperature control mechanism is to be independent of the materials inside the reactor and the reactions they undergo.

c. The set point (the desired temperature) for the reactor is constant through the experiment.

Note: In most cases the set point would change through the experiment, this implies that the system is dynamic and differential equations are needed to describe the system. The last assumption eliminates this requirement by considering the system as time independent, or in other words static.

C. Control Configuration

In most cases involving control of chemical process a mathematical model based on the theoretical approach of relating the variables that govern the process to each other is preferred. In this case modeling was done experimentally because of the availability of analog controllers that were implemented on the system but were not able to control the temperature inside the reactor. An oscillation of 50°C to 100°C resulted when the analog controllers were used.

Data were collected to help understand the system better and to study

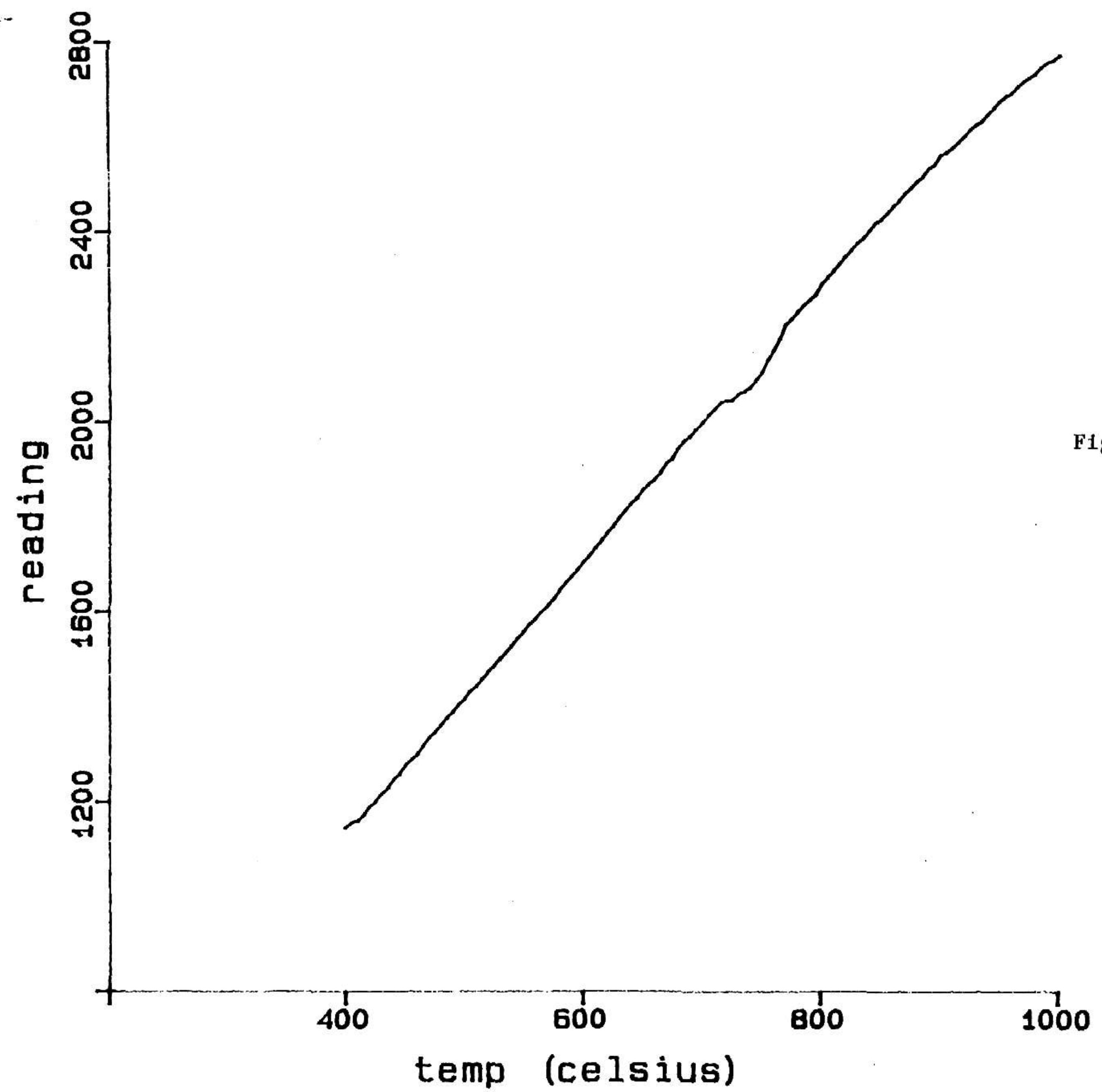


Fig. 1. Digital readings from A/D versus temperature. Gain of the differential amplifier is set to 180.

the behavior of temperature at different levels inside the reactor. Figure 1 shows a sample of these data. A set of equations that describe the temperature curve were derived (see software part) and two types of control were implemented.

a. Feedback control. The temperatures inside the reactor were sampled and compared with the desired temperature. Based on the result of this comparison the heaters were turned on or off. (See Figure 2.)

b. Feed forward control. It was learned that the temperature inside the reactor keeps on rising when reaching a set point even if the heaters were turned off. This rise was about 60°C to 100°C and requires around 30 minutes to bring the temperature back to the set point. Feed forward control was helpful in anticipating this rise and adjusting the desired temperature to avoid overshoot thus improving the steady state response.

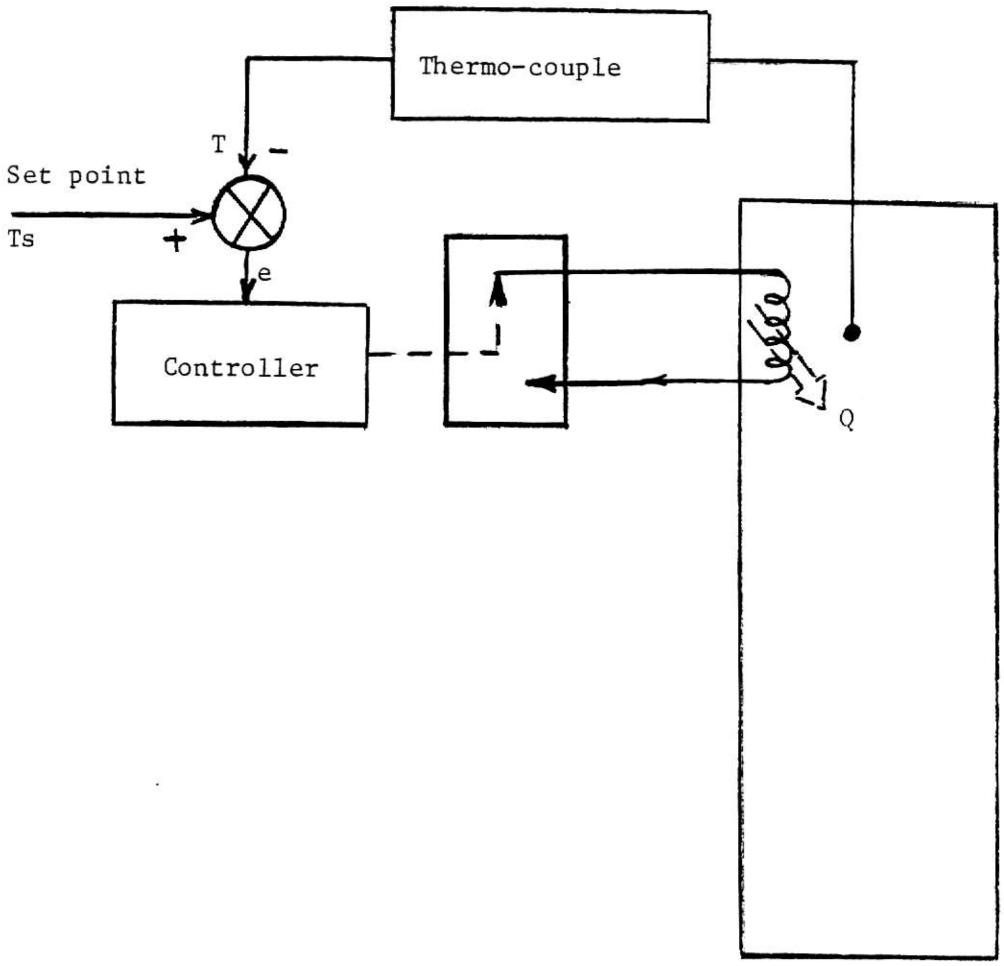


Fig. 2. Feed back temperature control for the reactor.

HARDWARE DESCRIPTION

The hardware of MBHC is divided into three major parts: The control unit, an input unit, and an output unit. The system block diagram is shown in Figure 3; the schematic diagram and list of I/O memory locations can be found in Appendix B.

A. Control Unit

The Commodore 64 is the brain of the whole system and controls all other parts of the MBHC. It sends signals to a command decoder which decodes these signals and presents them to the input and output units. The Commodore 64 sends control signals which depend upon the sampled inputs or the desired response of the reactor as specified by the programmer.

B. Computer Interface Signals

Several signals from the computer are used in the decoding process. These signals are:

IO1: Goes low each time any of the locations 56832 to 57087 in decimal, or DE00 to DEFF in hexadecimal are accessed.

Note: Twelve of these locations, 56832 to 56843 are used to send commands. Four more locations 56844 to 56847 are reserved for PIA.

$\phi 2$: The $\phi 2$ clock is a timing reference used to enable D1 and D2.

$R/\overline{W}$ Read/Write: The $R/\overline{W}$ signal is supplied by the microprocessor and controls the direction of data transfers of the 6526. A high on $R/\overline{W}$ indicates a read (data transfer out of the 6526), while a low indicates a write

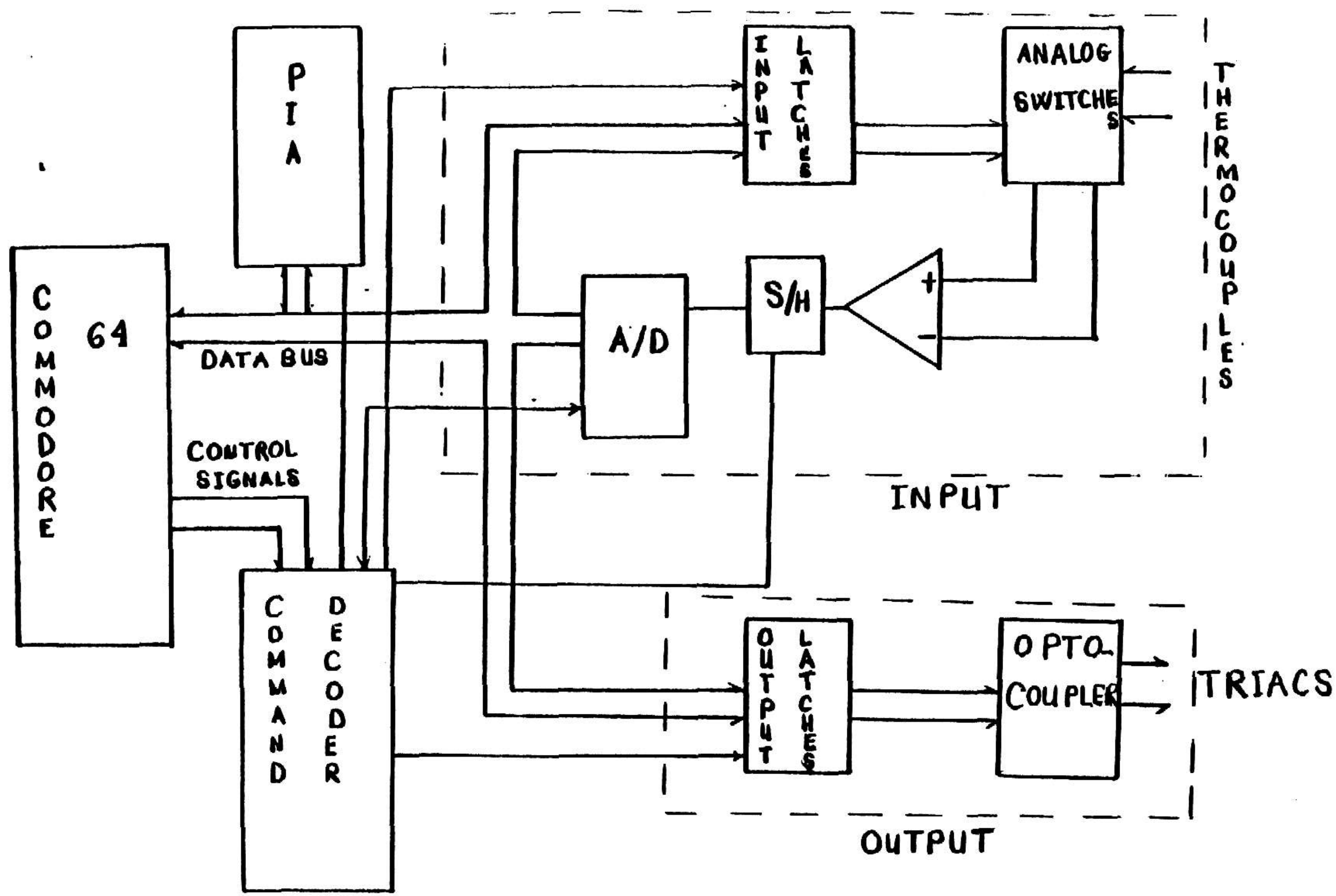


Figure 3. Block diagram

(data transfer into the 6526).

A_0-A_{15}: Address bus. Only A_0-A_3 are used. They select any one of 12 addresses starting from 56832.

D_0-D_7: Data bus Inputs/Outputs

The eight data bus lines transfer information between the computer and the system.

C. Command Decoder

The commands are decoded by two 74LS138 3-to-8 Line Decoders, D_1 and D_2. Their outputs are buffered by three 7404 Hex Inverters.

The role of D_1 is to control the analog switches connected to the thermocouples. D_1 is enabled when the following instruction is executed.

POKE Address, N

where N is a number used to select a specific channel of a switch, see Table 1. The address is one of four addresses from 56840 to 56843 where A_3 is high.

The role of D_2 is to control the triacs which are connected to the heaters, the sample and hold, and the analog to digital converter. D_2 is enabled when the following instructions are executed.

POKE Address, N (to control heaters); or

PEEK (Address), to read thermocouple

where N is a number which determines the command to be executed and address is one of eight addresses from 56832 to 56839 where in A_3 is high.

D. Commands

1. Select Thermocouple Channel, $\overline{LECX}$

Purpose: Select a channel in order to pass the voltage of the corresponding thermocouple through an analog switch to the analog to digital converter.

Instruction: POKE 56840,17

Sequence of steps: The execution of the above instruction triggers the following:

A_3 is buffered and inverted by B_2 to give $\bar{A}_3$

If $\bar{A}_3$ is low D_1 is enabled and y_o goes low.

If A_1 and A_o are low y_o buffered by B_1 makes $\overline{LEC1}$ high and LC1 is loaded from the bus with 00010001 (17 decimal).

The outputs of LC1 enable analog switches CH1H and CH1L. Thus thermocouple 1 is connected to the differential amplifier AD521 which amplifies the difference voltage 180 times (Gain is set to 180 by external circuitry) and passes the output to LF398A S/H sample and hold.

Note: To close any other channel the same sequence occurs with a different address and different data. Refer to table one for a list of these addresses and data.

2. Start conversion, CE

Purpose: To start A/D conversion of the analog voltage held by the S/H.

Note: A/D is operated under "stand alone" mode.

Instruction: Peek (56837) (ignore input)

Sequence of steps:

D_2 is enabled, Y_5 goes low

Y_5 gets buffered by B_3 and CE goes high

Conversion starts when R/$\overline{C}$ on A/D goes low.

3. Read status bit, STS

Purpose: To find out if the A/D has finished conversion. i.e. when STS output of A/D goes low.

Instruction: If Peek (56832) >=128, (then conversion is done)

Sequence of steps:

D_2 is enabled, Y_1 goes low

Y_1 gets buffered by B_2 and 74126 tri-state is enabled to pass the STS bit to D_7 of the data bus. (conversion done if $D_7 = 0$).

4. Hold , Q_2 SET

Purpose: puts S/H in hold state to provide a stable signal for the A/D for accurate conversion.

Note: The hold capacitor is chosen to be 0.02 μF to provide a holding time of approximately 9o to 100 μsec. Maximum conversion time of the A/D is 35 μsec.

Instruction: Poke 56839, X   X is any number

Sequence of Steps:

D_2 is enabled, Y_7 goes low

Y_7 buffered by B_3, makes PR2 goes high

Q_2 of 7476 goes low (clear is high)

S/H is put in hold state.

5. Read output of A/D, $\overline{OEH}$ and $\overline{OEL}$

Purpose: Get the binary number corresponding to the analog voltage of the thermocouple into the computer.

Instruction: Num = [Peek(56835)*16] + [Peek (56836)]
A B

Sequence of steps:

Step A

D_2 is enabled, Y_3 goes low

Y_3 is buffered by B_2 and $\overline{OEH}$ goes high and L_1 is enabled

74LS245 passes data from L_1 to computer

Step B

D_2 is enabled, Y_4 goes low

Y_4 is buffered by B_2 and $\overline{OEL}$ goes high and L_2 is enabled.

74LS245 passes data from L_2 to computer

$\overline{OEL}$ is also connected to clear 7476 flip-flops and flip-flops are set.

Q_1 lights LED to indicate data transmission.

Q_2 goes high and puts sample and hold in sample state.

6. Heaters On/Off, $\overline{LEHX}$

Purpose: To turn heaters on or off by firing triacs. To turn a heater on/off a triac has to be fired by providing a low/high signal to a MOC3010 opto-triac.

Instruction: Poke 56832, 255

Sequence of steps: The instruction above is chosen as an example to turn on H_1 (heater one). The execution of the above instruction triggers the following:

D_2 is enabled, Y_o goes low

Y_o is buffered by B_2 and $\overline{LEH1}$ goes high

LH1 is loaded from the bus with 1111 1110 (255 decimal).

Then the data are passed to MOC 3010's

MOC 3010 #1 gets a low signal which fires its corresponding triac and turns H_1 on.

Note: A PIA peripheral interface adaptor is added to the system, for possible future expansion, which gives the system the capability of interfacing with other devices. For more details on pin out of the PIA refer to table three Appendix B.

SOFTWARE DESCRIPTION

Four programs were written to operate the MBHC. It should be noted that these programs are typical examples of how to operate the MBHC; the user may add his own programs to achieve his goals. It is hoped that these four programs will give the user a firm idea about the capability of the hardware of MBHC. The programs included in this paper are the "Pre-timed Control" which differs from the "Control Program" only in the initialization part which allows the user to present a defined time of operation. The "Heaters On" is a small program to set the heaters on or off and frees the computer for other work; and finally the "Performance Test" which checks the hardware and gives the user messages about defective parts, as an aid maintenance of the circuit. In the rest of this section, a detailed description of each program is given. Also, listings and flow charts of all programs are available in Appendix C.

A. Control Program

This program implements the control process which aims at stabilizing the temperatures inside the reactor at all four levels. The user enters "little" or "big" and the desired temperatures at each level of the reactor. The program stores the user's information and goes to work; it starts by closing the switch corresponding to the thermocouple monitoring the temperature at each level; after the signal is passed, it gets amplified at the differential amplifier, the sample and hold is put in hold state and then the A/D is enabled. The binary signal is read and then the temperature that corresponds to this number is found by comparing it to a set of linear equations that have been stored in memory. These equations were obtained by dividing the curve in figure 1 into small linear portions and writing linear equations to represent them. After the temperature is found, the program compares it to the set point, and if it is greater or less the heater corresponding to that level is turned off or on. The program does the same thing for each level and repeats the whole process for all levels ten times and then averages the temperatures and displays them to the user.

Feed forward is implemented in the program by subtracting 50°C from the set point as entered by the user to obtain a temporary set point. This gets incremented gradually when reached by a ten cycle average until the entered set point is reached.

The program allows eight extra channels to be read and displayed. Four parts that form the control program; 1) initialization 2) display subroutines, 3) main program, and 4) the temperature calculations are described below.

1. Initialization. In the beginning, all the analog switches are opened. Then the heaters are set to off state. After that, the program variables are initialized to provide proper initial conditions. Finally, the desired temperatures are set by the user by answering program prompting messages.

2. Display Subroutine. This subroutine clears the screen and sets the display format for displaying the current temperature readings of the operating channels.

3. Main program. This program initializes its variables and closes a switch for a temperature reading, and sets the sample and hold in the hold state. Then the A/D starts its conversion. The number read from the A/D is

used to find a linear equation for the temperature calculation. Then, the calculated temperature is compared to its desired temperature and based on this comparison turns heaters on or off. Note, since there are two reactors utilizing first eight channels, the calculated temperatures of the extra operating channels, nine to sixteen, are displayed only. Due to system steady state operation, the program displays average of ten consecutive readings for each specific channel.

4. Temperature Equations. Since A/D output is not proportional to temperature as measured by a sliding thermocouple, readings were made to determine linear equations which could be used to find the temperature from 0° to 1100°C. The number read from the A/D is compared with its references. Specific reference numbers corresponds to a linear equation which is used to find the final temperature reading.

B. Pretimed control Program

This program is similar to the Control program except the initialization part. The purpose of this program is to be able to operate the MBHC at off hours by setting the desired type of operation and giving the starting time at which the system is desired to begin its work. The microcomputer will delay the starting until the desired time is reached by using its timer.

C. Heaters On/Off

This program is used to set the reactor's heaters on or off and frees the microcomputer to the user. This program is useful when the user needs to work with the microcomputer for a different purpose until the working temperatures are reached.

D. Performance Test

It is important to facilitate maintenance and provide repair clues to the user. This program concentrates mainly on the analog switches which could be a probable source of error. It checks the analog switches one at a time and points out the defective ones that need to be replaced. It also checks the optocouplers in conjunction with the triacs and the defective ones can be replaced. This program along with the hardware schematics will help in replacing defective parts. It works interactively. Prompting the user through displayed messages to perform the necessary steps to conduct the diagnosis. This test consists of opening all the switches and measuring the output voltage to the differential amplifier which should read zero. If not, one or more switches are defective so the output voltage of each switch is checked alone to screen out the defective ones. A similar test is conducted on the output part concerning the triacs and the defective one is picked up.

FUTURE EXPANSION

MBHC can be expanded to accommodate new features. The user can use MBHC in a larger profile on two fronts. With respect to the software, the system is very flexible and can be programmed to meet the requirements of new jobs. The computer and the disc drive provide large memory space for software expansion. Regarding hardware, MBHC can be connected to other devices such as printer, tape recorders, and other microcomputers. The system can be linked to these devices via the peripheral interface adapter, PIA, (the pin-out is given in Appendix B). Also, one output latch provides a means to expand the output section of MBHC to more triacs which means another reactor or eight new heaters.

PROCEDURE OF OPERATION

1. Important: All power to the systems should be off before starting.

2. Connect the thermocouples to correspondingly labeled yellow connectors.

3. Make sure that the connecting wires from the thermocouples are hooked correctly to the back of the interface box. Check the diagram in Appendix B for the labeling of the connectors at the back of the black box.

4. Turn on the power in the following order: a. TV, b. Disc drive, c. DC power supply, d. Microcomputer, e. AC power to the reactors.

5. Load the program for the desired operation.

6. Run the program.

7. Follow the prompts messages of the program.

8. Stop the program when done.

9. While shutting off the system the microcomputer should be turned off first.

CAUTION

1. The above steps should be followed in order to avoid damage to the circuit.

2. To avoid damage to the analog switches, do not take out the connecting wires from the thermocouples while power is on.

REFERENCES

1. Richard A. Pearman, Power Electronics, Solid State Motor Control, Reston Publishing Company, 1979.

2. George Stephanopoulos, Chemical Process Control. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.

3. Texas Instruments, Inc., The TTL Data Book for Design Engineers, Second Edition, 1979.

4. Analog Devices, Inc., Data Acquisition Products Catalog Supplement, 1979.

5. National Semiconductor Corporation, CMOS Databook, 1981.

6. Data Manual, Motorola Microprocessors, 1981.

7. Omega, Complete Temperature Measurement Handbook and Encyclopedia, 1984.

8. National Semiconductor Corporation, Linear Databook, 1978.

Appendix A
Description of the reactor and the chemical procedure.

Note:

The information in this Appendix was provided by Mr. Neogi Debashis from the Chemical Engineering Department. It gives a description of the reactor used in the project and the procedure that was followed using the old controllers. This appendix explains the chemical aspects of the system and its applications.

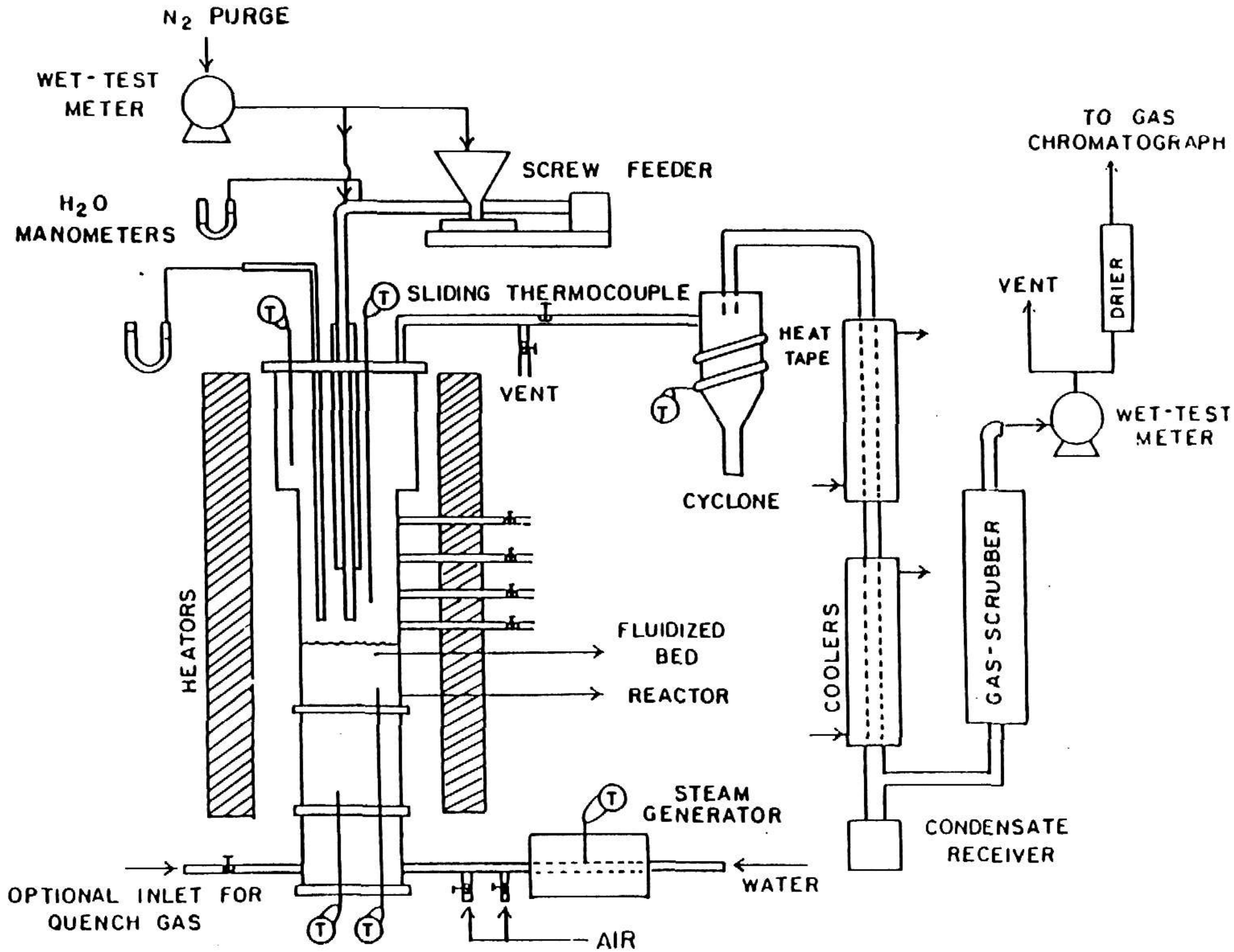


Fig. 4. Bench scale fluidized bed coal gasification system.

Facilities

The fluidized-bed reactor used in this work was designed to study the pyrolysis and gasification of various carbonaceous materials, e.g., coal, at atmospheric pressure and over a temperature range from 650 K to 1450 K. The experimental system consisted of three sections (Figure 4); the reactor, gas clean-up and gas sampling sections. The reactor consisted of four zones (Figure 5); the disengaging, fluidized-bed, packed-bed and gas-inlet zones. Inconel 600 alloy was used to construct the reactor because of its high temperature resistance and capacity to withstand rapid heating and cooling.

The reactor was constructed from a 10.16 cm I.D. by 55 cm length schedule 40 pipe. It was fitted with a pipe, 15.24 cm I.D. by 20 cm length, of the same material for the upper section; this section served as the disengaging zone (freeboard) of the reactor. The botton 25 cm served as the gas distribution and gas preheater zone. The top 15 cm of this zone was packed with aluminum oxide pellets (0.5 cm in diameter). The packed-bed section allowed the fluidizing gas to uniformly enter the fluidized bed zone. A thermocouple was placed in the inlet zone to monitor the fluidizing gas temperature. This zone was separated from the packed bed zone by a 60 mesh 316 SS screen (opening width = 0.23 mm, open area = 30.5%). The packed-bed and fluidized-bed zones were also separated by an identical screen in the form of an inverted cup. The screen was held in place, from the top, by six metal studs, about 0.635 cm long welded at equal distance on the inner wall of the reactor, and by the packed bed from below. This screen prevented the percolation of the bed material into the packed bed zone. Instead of gaskets, heat resisting sealant was used between all the flanges.

The inert matrix of the bed contained 25% by weight of limestone and 75% by weight of silica sand; the limestone was for preventing agglomeration of particles which typically occurred in a bed composed only of sand (Walawender et al., 1981, Fan and Walawender, 1983). The size of limestone particles ranged from -7 to +50 mesh (2.82 mm - 0.287 mm), and thatof sand particles from -30 to +50 mesh (0.59 mm to 0.287 mm). The static-bed height was 8 10 cm, and the expanded bed height was 12 14 cm.

The reactor was heated by means of ten quarter cylindrical electrical resistance heaters, each capable of delivering up to 1200 watts of power with a maximum sustained operating temperature of 1550 K. Five heaters were for the upper part of the reactor and five for the bottom part, thus forming two distinct sets of heaters. Voltage to each set of heaters was controlled by two PID controllers (Omega model 49K-814). There were five thermocouples installed in the reactor. These thermocouples were of the chromel-alumel type with 1/8 inch inconel sheaths. One of them, a sliding thermocouple, was used to measure the temperature profile inside the reactor. The others were located at the freeboard, fluidized-bed section, preheating zone, and the middle of the reactor. The controllers recorded the temperatures from these thermocouples and accordingly activated the heaters as required to maintain the temperature inside the reactor at a preset value. A pressure probe, connected to a manometer, indicated the bed pressure and the state of fluidization.

The feed was introduced into the reactor by gravity flow through a vertical feedpipe (3 cm I.D.) which discharged at a location about 8 cm above the static bed. A Vibra Screw Feeder (Model SCR-20), with a solid core flight screw, fed the coal particles at an uniform volumetric flowrate. The screw drive was designed with a closed loop feedback system, which immediately compensated for variations between the actual speed and the set point. A purge flow of N_2 aided solid flow through the feedpipe and prevented gas backflow and subsequent condensation of vapor in the feeder. To prevent the feed materials from prematurely devolatizing before it reached the bed, the feedpipe was equipped with a water jacket which maintained the temperature inside the feedpipe below 400 K.

Steam served as the fluidizing medium, and was produced externally in the Sussman Hot Shot electric boiler (Model MB-6) and was supplied to the gas preheater section at a temperature around 400 K and a pressure of 202.6 kPa. The gas exiting from the reactor was passed through a cyclone to remove the fine solid particles, e.g., char, from the gas stream. The cyclone was well insulated and maintained itself at 500 600 K. After leaving the cyclone the gas stream was passed through in series two water-cooled, single-pipe heat exchangers. This resulted in condensation of steam and tar, which was collected in a condensate receiver. Further gas cleaning was accomplished by means of a dry scrubber packed with glass wool. The scrubber was effective in removing the fine mist traveling with the gas without creating any appreciable pressure drop. A wet test meter connected to a strip chart recorder was used to measure the flow rate of the gas. A side draw of the off-gas was passed through a column packed with drierite ($CaSO_4$) and then sent to an on-line process gas chromatograph (Applied Automation Optichrome) for analysis. The nitrogen purge through the feed pipe was measured by a wet-test meter. The concentration of the nitrogen in the product gas along with its rate allowed computation of the product gas rate.

Procedure

For start-up each experimental run, the heaters and steam generators were turned on and the controllers were set at the desired operating temperatures. During the heat-up period, air served as the fluidizing agent and also as the feed pipe purge gas. The water flow to the jacketed feed-pipe was also initiated; to prevent overheating of the feed-pipe. The steam generator was operated in such a way that steam was supplied at constant pressure. Once steam became available, the fluidizing air flow was gradually replaced by steam. The volumetric flow rate of steam required to maintain fluidization was determined by collecting condensate, downstream from the heat exchangers and was controlled by a needle valve on the steam line.

When the bed and freeboard reached the selected operating temperatures, the axial temperature profile was measured and minor corrections made with the controllers to ensure a uniform profile throughout the reactor. The exit-gas from the system was analyzed to ensure that it was free of air. At this point, the system was ready for the feeding. The total start-up time from a cold start was about two hours and about one hour from a warm start. Normally the heaters were not turned off but just turned down at the end of each run, so that the system stayed warm overnight.

A slight drop in the temperature of the reactor occurred when feeding was initiated; this was automatically compensated by the controllers. A gas sample was taken about 5 minutes after feeding began and every 11 minutes thereafter. The flow rate of the condensate and that of nitrogen were measured every 10 minutes throughout the run. A typical experiment at a given temperature lasted 100 to 120 minutes with the last 50 and 60 minutes yielding steady gas chromatograph readings. The feed rate was evaluated by disconnecting the lower section of the feed pipe and weighing the effluent collected over three time intervals each lasting 3 minutes. This was done at the start and end of run.

It was impossible to measure the total char produced in the experiment because of the hold-up of char in the bed and connecting pipes. It was also impossible to measure the total tar produced due to lack of adequate separation facilities. Also, significant quantities of tar were held up in the heat exchangers. Consequently, overall material balances were not attempted.

Chemical Analyses

Analyses of the dry off-gas were conducted by an Applied Automation (Optichrom 2100) on-line process gas chromatograph. The components of interest were H_2, CO, CO_2, CH_4, C_2H_4, C_2H_6, C_3H_6, C_3H_8, O_2, and N_2. The chromatograph had a cycle time of 11 minutes. Moisture and ash analyses of feed material were performed according to standard ASTM procedures in a ventilated oven and muffle furnace respectively. Elemental analyses of the feed were carried out with a Perkin-Elmer model 240b elemental analyzer.

Operating Conditions

The operating conditions for all the experimental runs are summarized in Table 1. By adjusting the steam rate according to the temperature of the run, an attempt was made to maintain a fairly constant gas phase mean residence time for the volatiles over all the runs. The gas phase mean residence time ranged from 4 to 7 seconds and this was estimated on the basis of the reactor temperature, total dry gas flowrate, and steam rate.

The steam rate was varied from 15.33 g/min. to 11.0 g/min. The average feed rate varied between 4.4 g/min. and 5.8 g/min., even though the screw-feeder setting was the same for all the experimental runs. The principal experimental variable was the reactor temperature. All the experiments were performed with an uniform axial temperature profile throughout the reactor and freeboard.

Feed Material

The feed material for all the experiments was Bituminous coal from Rowe coal bed located in southeast Kansas. Pretreatment of the coal included hammer milling and size classification. The powder density of the coal was 1580 kg/cu. m. The Higher Heating Value (HHV) of the coal was 30.62 MJ/kg (calculated by the Dulong formula). The coal had a Free Swelling Index (FSI) of 5.5.

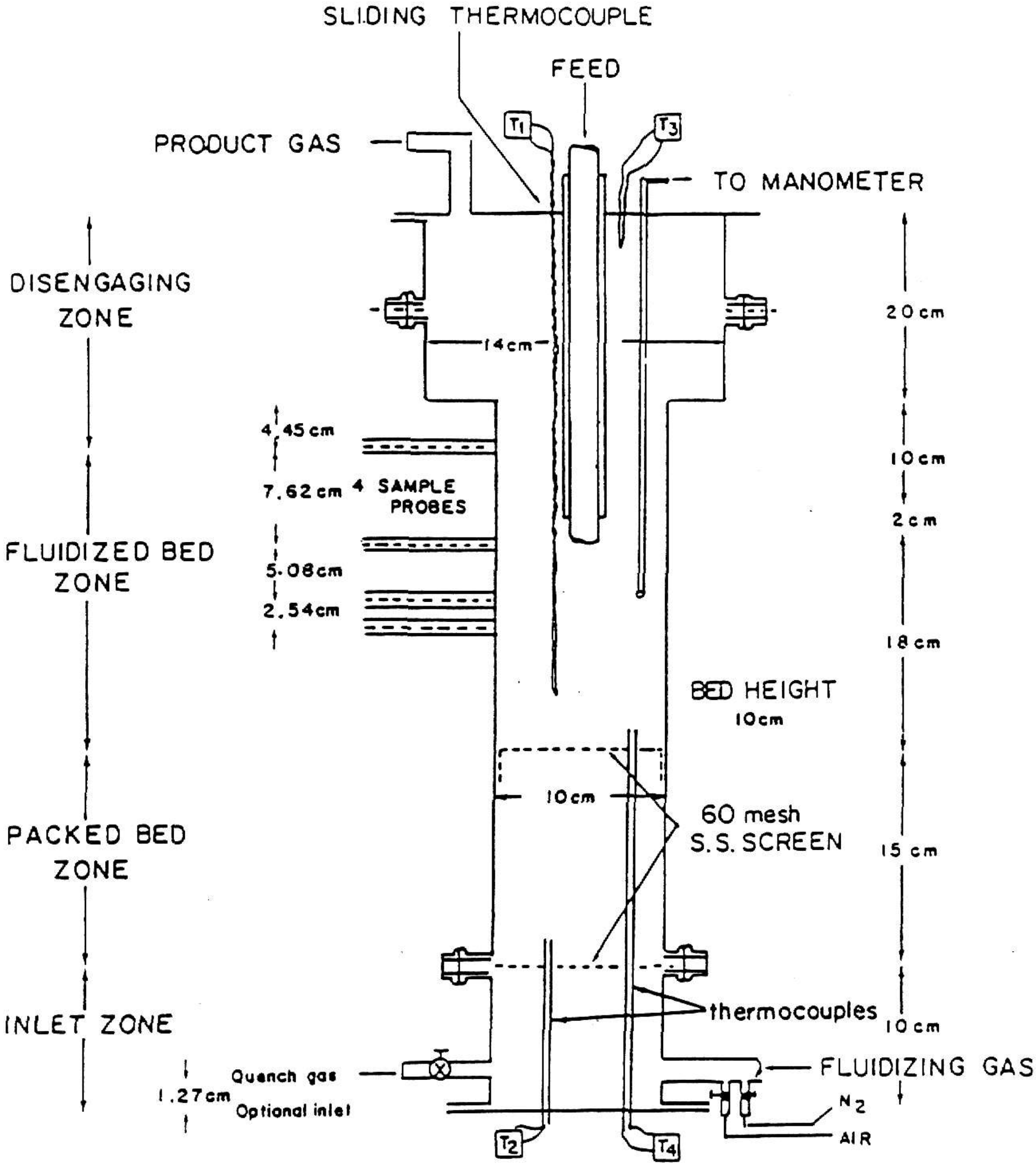


Fig. 5. Fluidized bed reactor.

APPENDIX B:
SCHEMATIC DIAGRAM
I/O MEMORY LOCATIONS
COMMODORE EXPANSION CONNECTOR
I/O CONNECTORS FORMAT
PIA PINOUT
LABELLING OF THE CONNECTORS AT THE BACK OF THE BLACK BOX

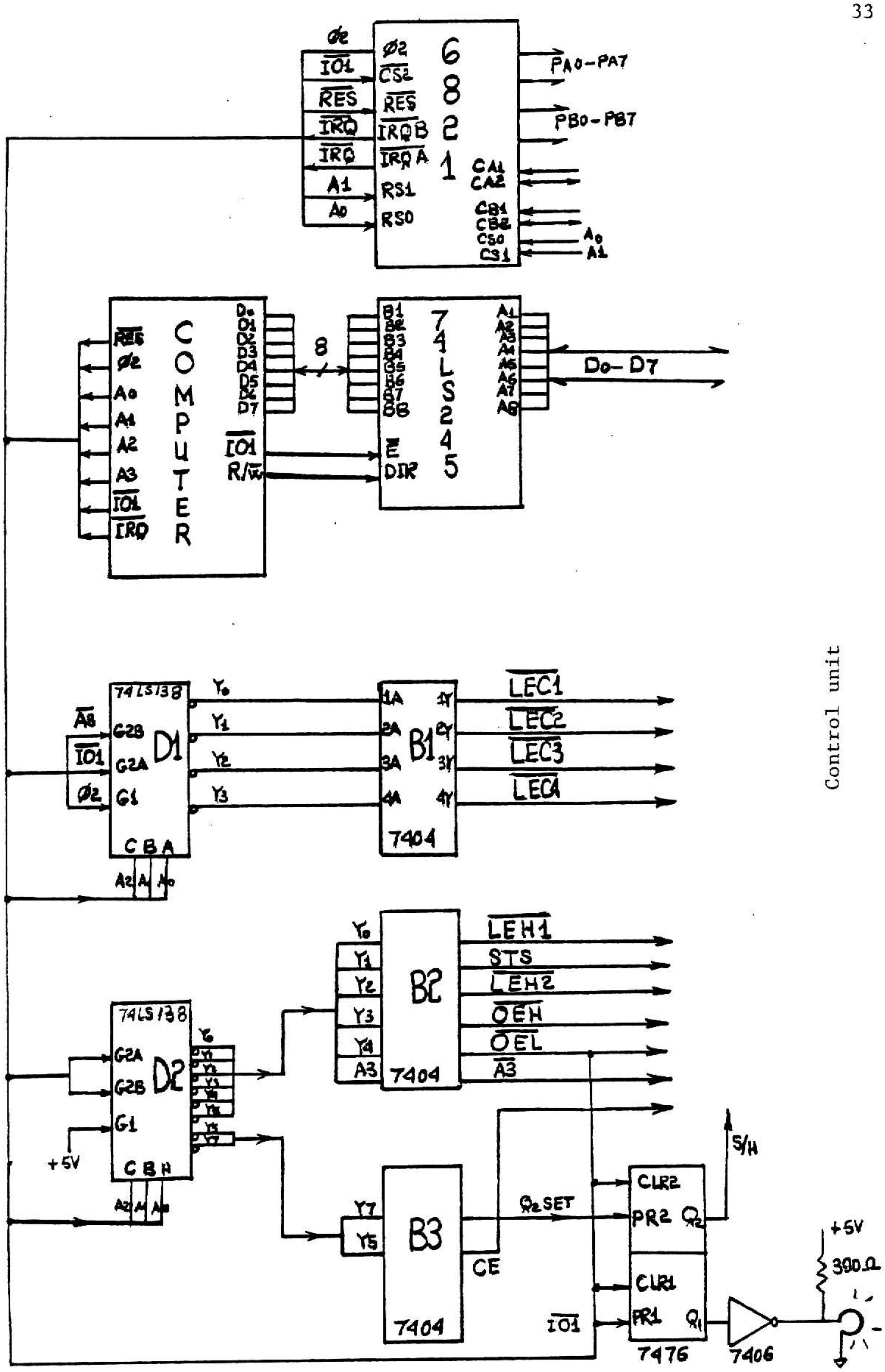
Schematic Diagram

In Order:

Control Unit

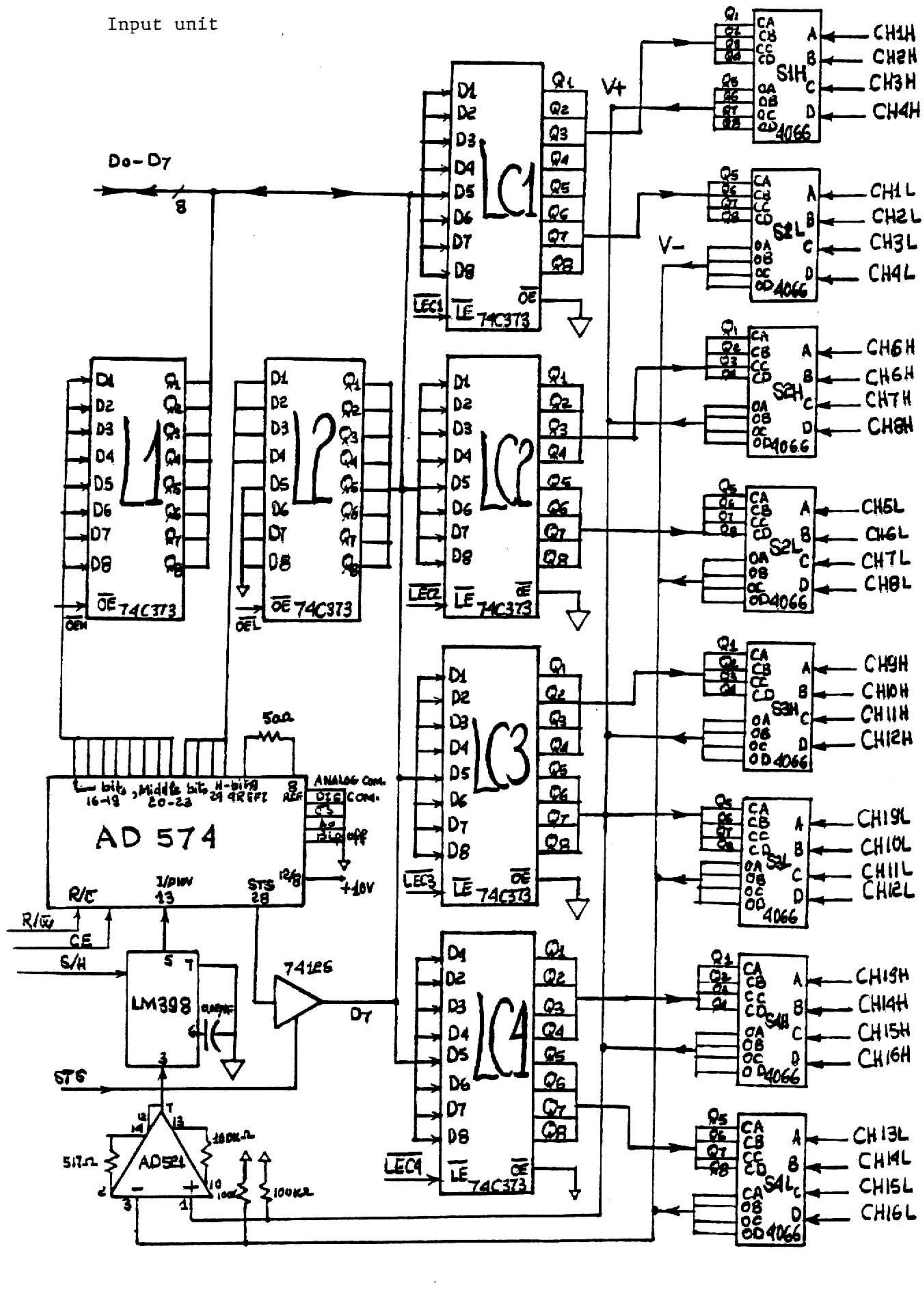
Input Unit

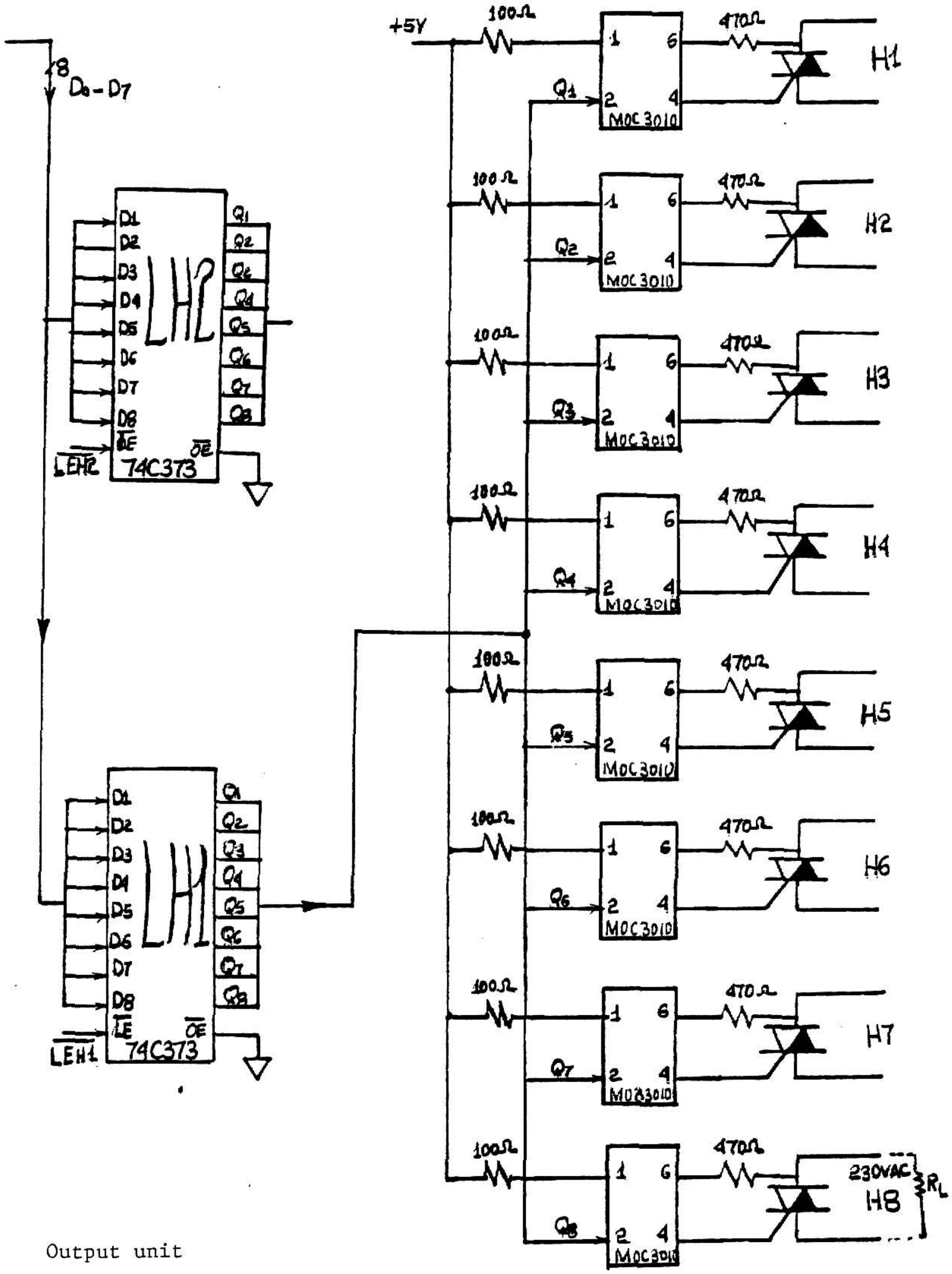
Output Unit



Control unit

Input unit


D0-D7
8
L1
74C373
L2
74C373
LC1
74C373
LC2
74C373
LC3
74C373
LC4
74C373
V+
V-
S1H
S1L
S2H
S2L
S3H
S3L
S4H
S4L
4066
CH1H
CH2H
CH3H
CH4H
CH1L
CH2L
CH3L
CH4L
AD 574
ANALOG COM.
DIG COM.
R/C
STS
28
+10V
50Ω
R/W
CE
S/H
LM398
STS
74126
D7
AD521
100KΩ



D0 - D7
8
+5V
100Ω
470Ω
D1
D2
D3
D4
D5
D6
D7
D8
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
LH2
LH1
74C373
LEH2
LEH1
LE
OE
MOC3010
H1
H2
H3
H4
H5
H6
H7
H8
230VAC
RL

Output unit

Input latch Address	Data	Channel #
56840	17	1
"	34	2
"	68	3
"	136	4
56841	17	5
"	34	6
"	68	7
"	136	8
56842	17	9
"	34	10
"	68	11
"	136	12
56843	17	13
"	34	14
"	68	15
"	136	16

Table 1: Input Channels memory locations and their corresponding data.

I/O MEMORY LOCATIONS

Every I/O chip on the interface circuit is enabled at a certain memory location by poke or peek statements. List of these memory locations and the enabled chips are available below in Table 2.

Memory Location	Enable Chip and Comments
56832	Output latch, control for firing the triacs
56833	Status Bit of A/D
56834	Output Latch, expansion control for extra triacs
56835	A/D data latch - bits 1 to 8
56836	A/D data latch - bits 9 to 12; also leaves the S/H in hold state
56837	Run A/D
56838	Unused
56839	Leaves the S/H im Sample State
56840*	Closes/opens Channels 1, 2, 3, 4
56841*	Closes/Opens Channels 5, 6, 7, 8
56842*	Closes/Opens Channels 9, 10, 11, 12
56843*	Closes/Opens Channels 13, 14, 15, 16
56844-47	PIA internal registers; refer to PIA data sheets in Appendix A.

Table 2. *More details in Table 1.

Commodore Expansion Connector

Commodore's expansion slot is connected to a 40 pin connector for interchanging I/O signals between the interface circuit and Commodore 64. Pin layout of this 40 pin connector is shown below:

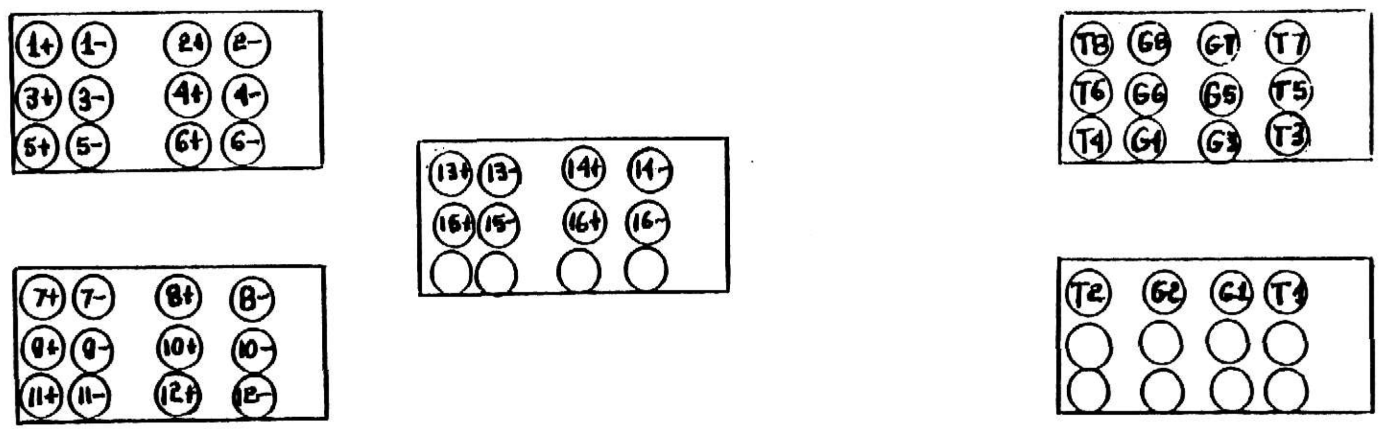
Pin #	Type	Pin #	Type
1	GND	21	GND
2	#5	22	A2
3	N.C.	23	A3
4	+5	24	A4
5	$\overline{IRQ}$	25	A5
6	$R/\overline{W}$	26	A6
7	Dot Check	27	A7
8	$\overline{I/O^{1}}$	28	A8
9	$\overline{I/O^{2}}$	29	A9
10	$\overline{ROML}$	30	A10
11	BA	31	A11
12	$\overline{DMA}$	32	A12
13	D7	33	A13
14	D6	34	A14
15	D5	35	A15
16	D4	36	SO2
17	D3	37	$\overline{NMI}$
18	D2	38	$\overline{RESET}$
19	D1	39	$\overline{ROMH}$
20	D0	40	GND

The gray and white wires from the expansion bus correspond to A1 and A0, respectively

I/O Connectors Format

The place for six female connectors is provided on the back of the interfaced box. Three of these connectors belong to input thermocouples which are directly connected to the input switches. The layout for thermocouples is shown in Figure 2. Since each thermocouple has two leads, one positive, and another negative special attention should be given for relocating or replacing thermocouples.

Another two connectors are used for firing the triacs. There are two wires for each triac which are labled Gate and Terminal, all Gate and Terminal wires are indicated by black and red wires, Figure 3 shows the layout format for triacs. Male connector pinout for PIA is given in Table 3. Refer to Appendix A for more information about PIA.



Thermocouples connectors

Triacs connectors

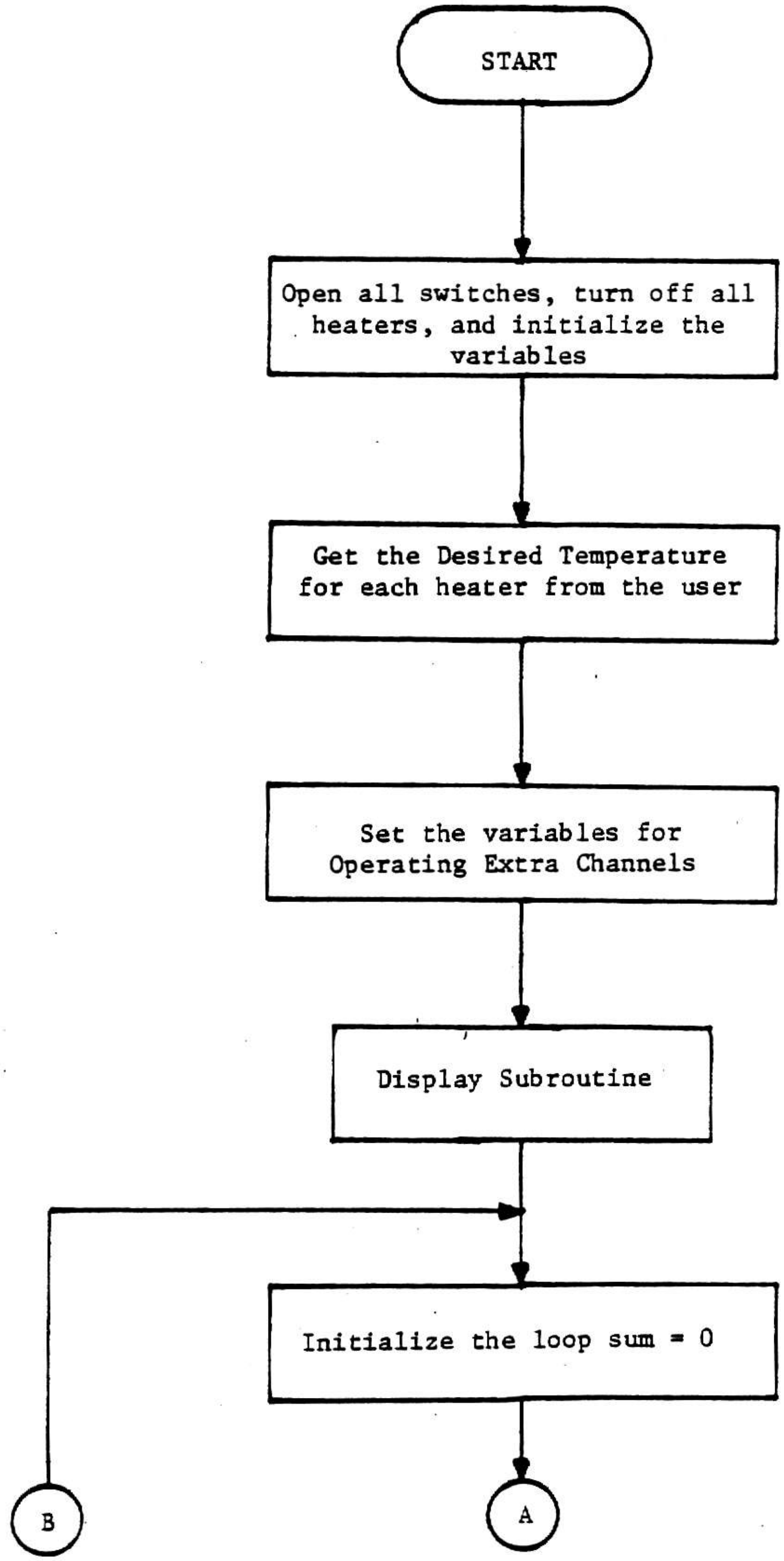
Figure 6. Connectors to the back of the box

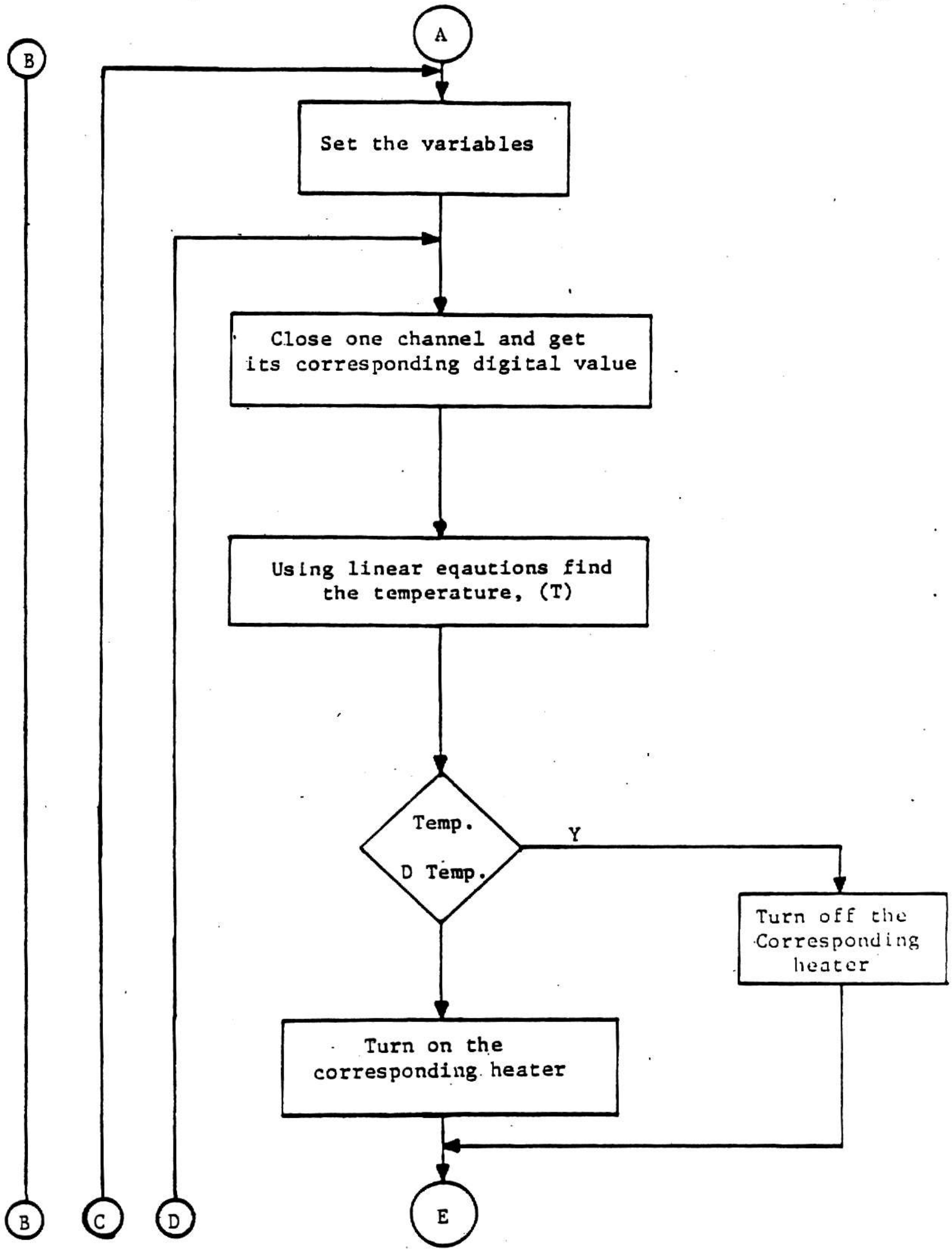
Pin #	Type	Pin #	Type
1	CA1	11	PB0
2	CA2	12	PB1
3	PA0	13	PB2
4	PA1	25	PB3
5	PA2	24	PB4
6	PA3	23	PB5
7	PA4	22	PB6
8	PA5	21	PB7
9	PA6	20	CB1
10	PA7	19	CB2

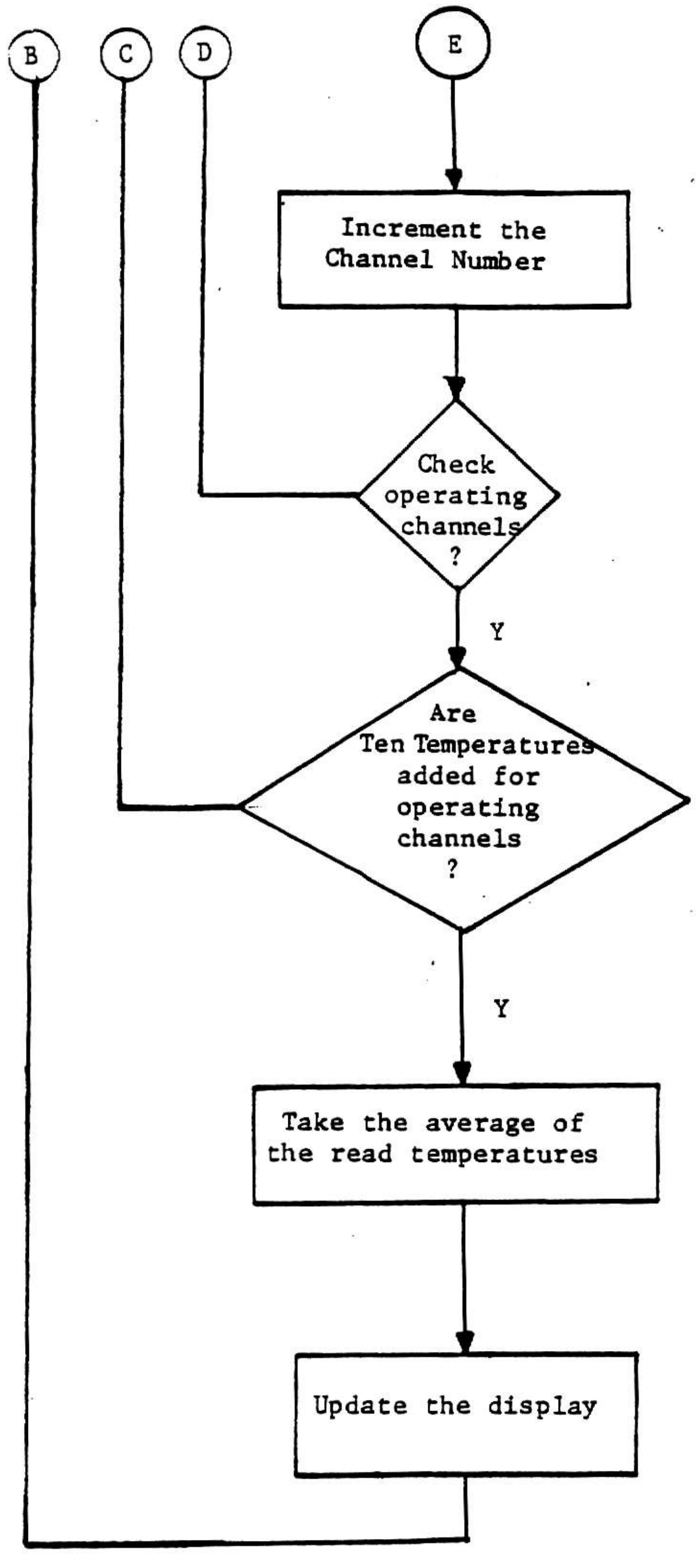
Table 3. PIA pin-layout

APPENDIX C: PROGRAMS LISTINGS AND FLOW CHARTS

CONTROL PROGRAM


START
Open all switches, turn off all heaters, and initialize the variables
Get the Desired Temperature for each heater from the user
Set the variables for Operating Extra Channels
Display Subroutine
Initialize the loop sum = 0
B
A


A
B
Set the variables
Close one channel and get its corresponding digital value
Using linear eqautions find the temperature, (T)
Temp.
D Temp.
Y
Turn off the Corresponding heater
Turn on the corresponding heater
B
C
D
E



Control Program Flowchart

```
ready.

3 rem control program
5 rem this prog. displays the desired tem. on screen also
6 rem $$$$$$ initialization part $$$$$
10 for i=56840 to 56844:rem open all switches
15 poke i,0
20 next i
21 rem oc is the bit pattern variable to set the optocouplers
24 rem j is an incremental variable
25 oc = 0   :j=0
30 poke 56832,oc
31 rem define arrays variables for desired temperatue dt,heaters' temp h
32 rem ef is extra flag used in case of extra channels operation
35 dim dt(8),h(16):ef=0
40 rem initi. 8 variables used for 8 extra channels
45 x=0:y=0:z=0:s=0:m=0:g=0:q=0:f=0
47 rem
48 rem set the desired temperatures
49 gosub 1000
50 print"do you want operate both reactors-y/n  "
55 input rr$
56 if rr$="y"then rd=1 :goto 65
59 gosub 1000
60 print "which reactor, small or big one"
62 input sb$
63 if sb$="big" then n=2: goto 65
64 n=1
65 if rd= 1  then n=0: goto 135
105 if n=1 then 135 :rem go to r1
110 input "what desired temp for heater# 5";dt(4)
111 gosub 1000: rem go clear the screen
115 gosub 1000
116 input "what desired temp for heater# 6";dt(5)
118 gosub 1000
120 input "what desired temp for heater# 7";dt(6)
121 gosub 1000: rem go clear the screen
125 input "what desired temp for heater# 8";dt(7)
126 gosub 1000: rem go clear the screen
130 goto 158 :rem jump to main prog
131 gosub 1000: rem go clear the screen
135 gosub 1000
136 input "what desired temp for heater# 1";dt(0)
137 gosub 1000: rem go clear the screen
140 gosub 1000
141 input "what desired temp for heater# 2";dt(1)
142 gosub 1000: rem go clear the screen
145 input "what desired temp for heater# 3";dt(2)
146 gosub 1000: rem go clear the screen
150 input "what desired temp for heater# 4";dt(3)
151 gosub 1000: rem go clear the screen
155 if rd= 1  then 110
158 print "do you want to use extra channels, y/n  "
159 input e$
160 if e$="y" then 950
163 gosub 900 :rem display sub.
164 rem set heaters' temps to zeros
165 gosub 1200
```

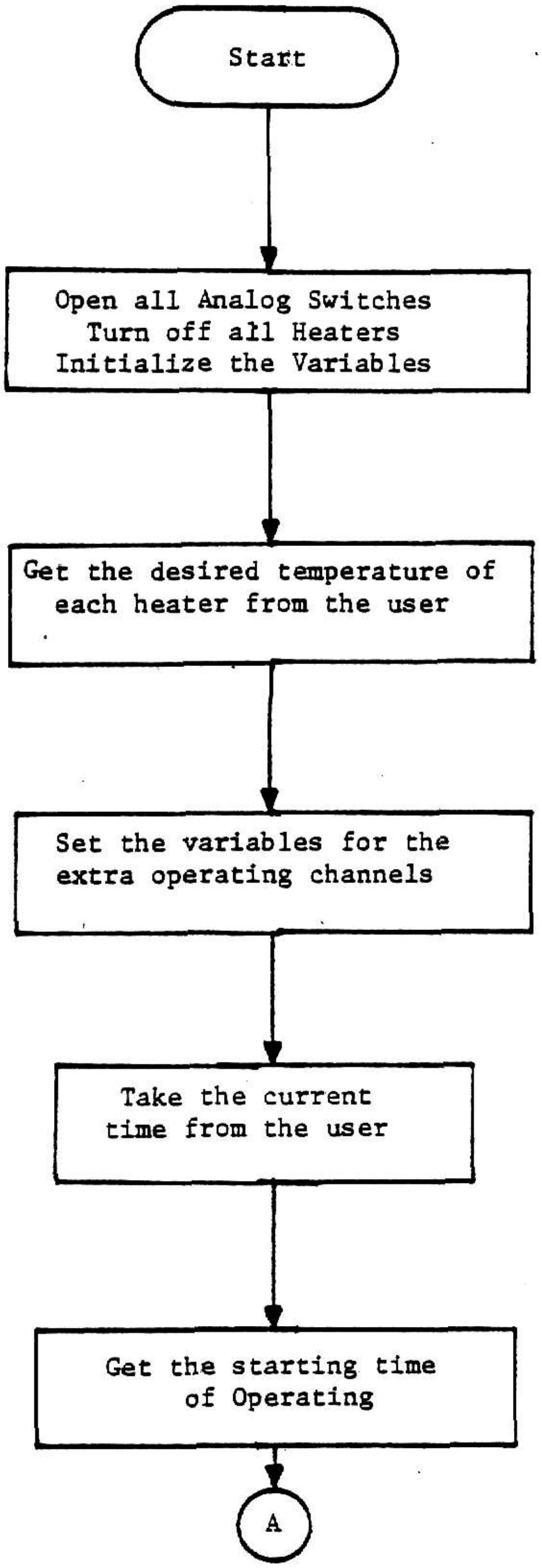
```
166 rem $$$$ main program $$$$$$
167 for a=0 to 15 :h(a)=0:next a
168 for  l= 1 to 10
169 ef=0
170 j=(n-1)*4 + 4*rd
173 for  p=(56839+n+rd) to (56839+n+2*rd)
174 for   i=0 to 3
176 for  a=56840 to 56843
179 poke   a,0 : next a
180 poke   p,(17*(2^i))
185 for a= 1 to 30 :next a
215 poke   56836,0 : rem s/h is in hold
220 poke   56837,0 : rem run a/d
225 if peek  (56833)>=128 then 225:rem read status
230 r=peek  (56835)*16 + peek  (56836)
235 poke 56839,0: rem s/h is in sample
245 if r<2638 then 460
250 if r<2715 then 465
255 if r<2789 then 470
260 if r<2862 then 475
265 if r<2934 then 480
270 if r<1731 then 400
275 if r<1812 then 405
280 if r<1886 then 410
285 if r<1959 then 415
290 if r<2037 then 420
295 if r<2088 then 425
300 if r<2189 then 430
305 if r<2262 then 435
310 if r<2339 then 440
315 if r<2413 then 445
320 if r<2488 then 450
325 if r<2563 then 455
330 if r<1343 then 370
335 if r<1421 then 380
340 if r<1499 then 385
350 if r<1576 then 390
355 if r<1652 then 395
360 goto 480
370 t=.31993*r + 46.005
371 goto550
380 t=.30633*r + 63.584
381 goto550
385 t=.33039*r  + 30.08
386 goto 550
390 t=.31843*r + 47.671
391 goto 550
395 t=.32451*r + 38.77
396 goto 550
400 t=.32035*r + 45.586
401 goto 550
405 t=.32337*r   +39.937
406 goto 550
410 t=.35183*r - 12.374
411 goto 550
415 t=.35285*r - 15.360
416 goto 550
420 t=.31201*r + 63.783
421 goto 550
425 t=.44970*r - 213.57
```

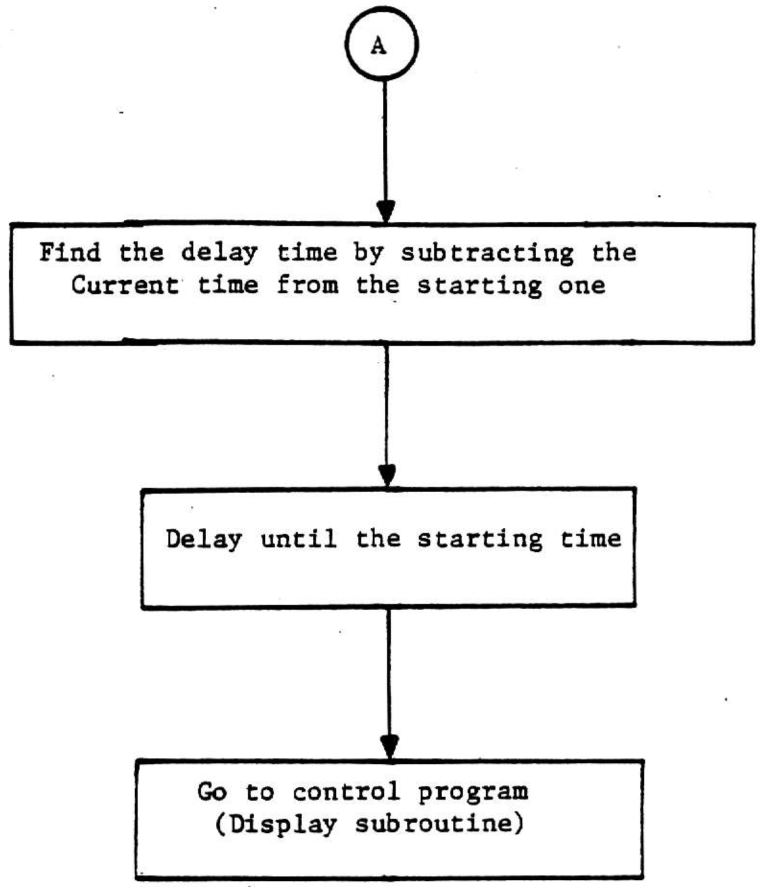
```
426 goto 550
430 t=.22206*r + 262.37
431 goto 550
435 t=.33649*r + 13.136
436 goto550
440 t=.32570*r + 37.783
441 goto 550
445 t=.33921*r + 6.1404
446 goto 550
450 t=.32827*r + 32.588
451 goto 550
455 t=.31418*r + 68.535
456 goto 550
460 t=.33741*r + 9.9699
461 goto 550
465 t=.33858*r + 6.3847
466 goto 550
470 t=.33856*r + 27.942
471 goto 550
475 t=.34695*r - 17.579
476 goto 550
480 t=.33070*r + 29.873
545 rem store temp readings in array h
550 h(i+j)=h(i+j)+t
555 if ef>0 then return : rem if extra channels needed then return
560 rem forward control routine
565 if ud=0 then 580
570 if t>=dt(i+j) then 1300
575 goto 625
580 if dl=0 then 600
585 if t>=dt(i+j) then dl=0:goto 600
590 goto625
597 rem
598 rem set the heaters on or off
600 if t< dt(i+j) then oc=(oc and (255-2↑(i+j))): poke 56832, oc  : goto625
620 oc=(oc or (2↑(i+j))):poke 56832,oc
625 next i
630 j=j+4
635 next p
637 rem check the extra, if used do it
640 if e$="n" then 695
645 if x>0 then i=0:p=56842:ef=1:j=8:gosub 176
650 if y>0 then i=1:ef=1:  gosub 176
655 if z>0 then i=2:ef=1: gosub 176
660 if s>0 then i=3:ef=1: gosub 176
665 if m>0 then i=0:p=56843:ef=1:j=13:   gosub 176
670 if g>0 then i=1:ef=1: gosub 176
675 if q>0 then i=2:ef=1: gosub 176
680 if f>0 then i=3:ef=1: gosub 176
685 rem all operating chan's were used
690 ef=0: j=0
695 next l
700 rem ********************************
705 rem  set and change the display
710 rem ********************************
720 print "[illegible]";
730 for a=0 to 7  step 2
740 h(a)=int(h(a)/10):h(a+1)=int(h(a+1)/10)
750 print tab(48)h(a);" ";tab(28)h(a+1);" ▒";
755 next a :rem next a
```

```
760 print "■■■■";
765 if e$="n" then 164
767 rem
770 for a=8 to 15 step 2
772 h(a)=int(h(a)/10):h(a+1)=int(h(a+1)/10)
775 print tab(50)h(a);" ";tab(30)h(a+1);" ■";
780 next a:
897 goto 166 :rem   loop again !
900 rem     ======================
902 rem      fix display format
903 rem     ======================
904 for a=1 to 50: print " ":next a
908 h$="current temperature readings"
909 ht$="heater":eq$="#":c$=","
910 ext$="extra operating channels"
915 print"■■■";tab(5)h$;tab(80)ht$;"1=";tab(14)c$;dt(0);tab(14)ht$;"2=";
916 print  tab(34)c$;dt(1);tab(45)eq$;"3=";tab(14)c$;dt(2);
917 rem
920 print tab(25)eq$;"4=";tab(34)c$;dt(3);
922 print tab(45)eq$;"5=";tab(14);c$;dt(4); tab(25)eq$;"6=";tab(34)c$;dt(5);
925 print tab(45)eq$;"7=";tab(14);c$;dt(6);tab(25)eq$;"8=";tab(34);c$;dt(7);
926 rem
927 print "■";tab(88)ext$;
930 print tab(87)eq$;"9=";tab(26)eq$;"10=";tab(86)eq$;"11=";tab(26)eq$;"12=";
932 rem
935 print tab(86)eq$;"13=";tab(26)eq$;"14=";tab(86)eq$;"15=";tab(26)eq$;"16=";
940 return  : rem back to action
950 rem exttra initi. part
954 gosub 1000
955 input"are you using ch # 9";wh$
956 if wh$="y" then x=1
960 input"are you using ch # 10 ";wh$
961 if wh$="y" then y=1
962 gosub 1000
965 input"are you using ch # 11";wh$
966 if wh$="y" then z=1
967 gosub 1000
970 input"are you using ch # 12";wh$
971 if wh$="y" then s=1
972 gosub 1000
975 input"are you using ch # 13";wh$
976 if wh$="y" then m=1
977 gosub 1000
980 input"are you using ch # 14";wh$
981 if wh$="y" then g=1
982 gosub 1000
985 input"are you using ch # 15";wh$
986 if wh$="y" then q=1
988 gosub 1000
990 input"are you using ch # 16";wh$
991 if wh$="y" then f=1
992 gosub 1000
995 goto163
1000 print "■"
1005 for a=1 to 44
1010 print "
1020 next a
1025 print "■■■■■■■■■■■":return
1100 rem #### undershoot set routine ##
1200 for a=0 to 7
```

```
 1210 dt(a)=dt(a)-60
 1220 next a
 1230 ud=1 :rem set delay variable
 1240 goto 166 :rem goto main program
 1300 rem delay until expected desired temperature
 1310 poke 56832,255:rem heaters off
 1320 for a=0 to 7
 1330 dt(a)=dt(a)+60
 1340 nexta
 1350 dl=1: rem set delay flag
 1360 ud=0
 1370 goto 166 :rem goto main program
ready.
```

MISSING PAGE


Start
Open all Analog Switches
Turn off all Heaters
Initialize the Variables
Get the desired temperature of
each heater from the user
Set the variables for the
extra operating channels
Take the current
time from the user
Get the starting time
of Operating
A


A
Find the delay time by subtracting the Current time from the starting one
Delay until the starting time
Go to control program (Display subroutine)

```
ready.

 3 rem pre-timed
 5 rem pre-timed control program without desired temperatures displayed
 6 rem $$$$$$ initialization part $$$$$
 10 for i=56840 to 56844:rem open all switches
 15 poke i,0
 20 next i
 21 rem oc is the bit pattern variable to set the optocouplers
 24 rem j is an incremental variable
 25 oc = 255 :j=0
 30 poke 56832,oc
 31 rem define arrays variables for desired temperatue dt,heaters' temp h
 32 rem ef is extra flag used in case of extra channels operation
 35 dim dt(8),h(16):ef=0
 40 rem initi. 8 variables used for 8 extra channels
 45 x=0:y=0:z=0:s=0:m=0:g=0:q=0:f=0
 47 rem
 48 rem set the desired temeratures
 49 gosub 1000
 50 print"do you want operate both reactors-y/n  "
 55 input rr$
 56 if rr$="y"then rd=1 :goto 65
 59 gosub 1000
 60 print "which reactor, small or big one"
 62 input sb$
 63 if sb$="big" then n=2: goto 65
 64 n=1
 65 if rd= 1  then n=0: goto 135
 105 if n=1 then 135 :rem go to r1
 110 input "what desired temp for heater# 5";dt(4)
 111 gosub 1000: rem go clear the screen
 115 gosub 1000
 116 input "what desired temp for heater# 6";dt(5)
 118 gosub 1000
 120 input "what desired temp for heater# 7";dt(6)
 121 gosub 1000: rem go clear the screen
 125 input "what desired temp for heater# 8";dt(7)
 126 gosub 1000: rem go clear the screen
 130 goto 158 :rem jump to main prog
 131 gosub 1000: rem go clear the screen
 135 gosub 1000
 136 input "what desired temp for heater# 1";dt(0)
 137 gosub 1000: rem go clear the screen
 140 gosub 1000
 141 input "what desired temp for heater# 2";dt(1)
 142 gosub 1000: rem go clear the screen
 145 input "what desired temp for heater# 3";dt(2)
 146 gosub 1000: rem go clear the screen
 150 input "what desired temp for heater# 4";dt(3)
 151 gosub 1000: rem go clear the screen
 155 if rd= 1  then 110
 158 print "do you want to use extra channels, y/n  "
 159 input e$
 160 if e$="y" then 950
 161 gosub 1500
 163 gosub 900 :rem display sub.
 164 rem set heaters' temps to zeros
```

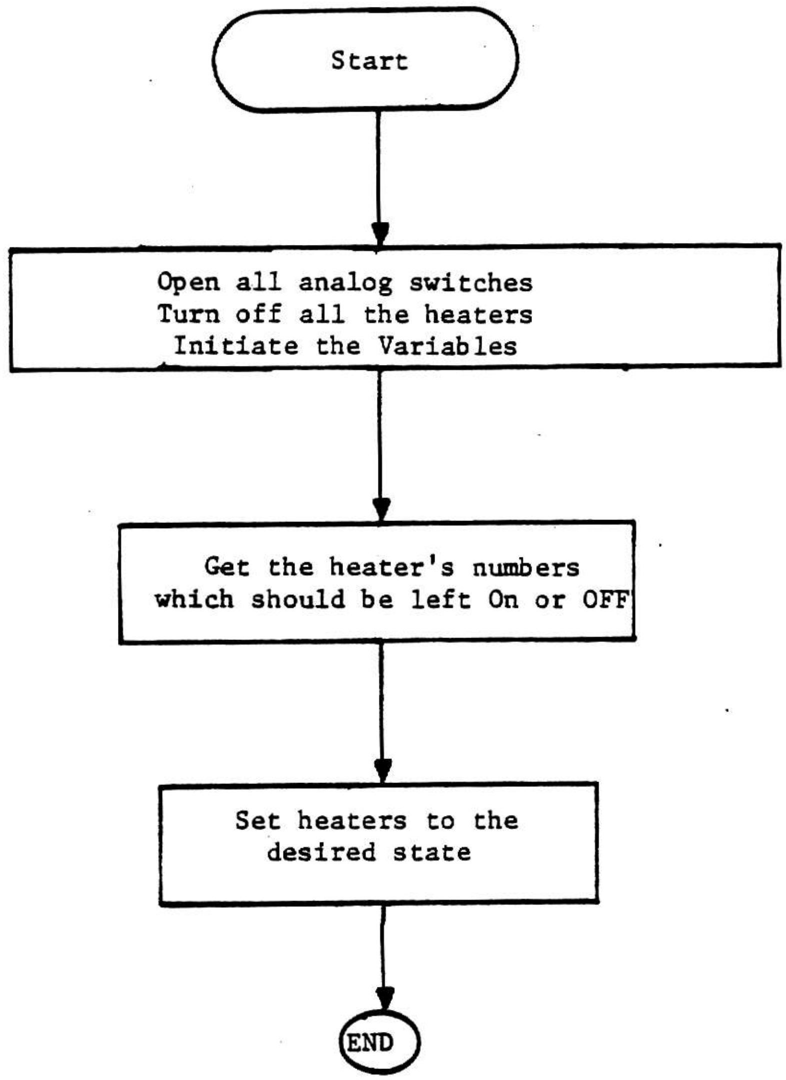
```
165 gosub 1200
166 rem $$$$ main program $$$$$$
167 for a=0 to 15 :h(a)=0:next a
168 for  l= 1 to 10
169 ef=0
170 j=(n-1)*4 + 4*rd
173 for  p=(56839+n+rd) to (56839+n+2*rd)
174 for   i=0 to 3
176 for  a=56840 to 56843
179 poke   a,0 : next a
180 poke   p,(17*(2%i))
185 for a= 1 to 30 :next a
215 poke   56836,0 : rem s/h is in hold
220 poke   56837,0 : rem run a/d
225 if peek  (56833)>=128 then 225:rem read status
230 r=peek  (56835)*16 + peek  (56836)
235 poke 56839,0: rem s/h is in sample
245 if r<2638 then 460
250 if r<2715 then 465
255 if r<2789 then 470
260 if r<2862 then 475
265 if r<2934 then 480
270 if r<1731 then 400
275 if r<1812 then 405
280 if r<1886 then 410
285 if r<1959 then 415
290 if r<2037 then 420
295 if r<2088 then 425
300 if r<2189 then 430
305 if r<2262 then 435
310 if r<2339 then 440
315 if r<2413 then 445
320 if r<2488 then 450
325 if r<2563 then 455
330 if r<1343 then 370
335 if r<1421 then 380
340 if r<1499 then 385
350 if r<1576 then 390
355 if r<1652 then 395
360 goto 400
370 t=.31993*r + 46.005
371 goto550
380 t=.30633*r + 63.584
381 goto550
385 t=.33039*r  + 30.08
386 goto 550
390 t=.31843*r + 47.671
391 goto 550
395 t=.32451*r + 38.77
396 goto 550
400 t=.32035*r + 45.586
401 goto 550
405 t=.32337*r   +39.937
406 goto 550
410 t=.35183*r - 12.374
411 goto 550
415 t=.35285*r - 15.360
416 goto 550
420 t=.31201*r + 63.783
421 goto 550
```

```
425 t=.44978*r - 213.57
426 goto 550
430 t=.22206*r + 262.37
431 goto 550
435 t=.33649*r + 13.136
436 goto550
440 t=.32570*r + 37.783
441 goto 550
445 t=.33921*r + 6.1404
446 goto 550
450 t=.32827*r + 32.588
451 goto 550
455 t=.31418*r + 68.535
456 goto 550
460 t=.33741*r + 9.9699
461 goto 550
465 t=.33858*r + 6.3847
466 goto 550
470 t=.33056*r + 27.942
471 goto 550
475 t=.34695*r - 17.579
476 goto 550
480 t=.33070*r + 29.873
545 rem store temp readings in array h
550 h(i+j)=h(i+j)+t
555 if ef>0 then return : rem if extra channels needed then return
560 rem forward control routine
565 if ud=0 then 580
570 if t>=dt(i+j) then 1300
575 goto 625
580 if dl=0 then 600
585 if t>=dt(i+j) then dl=0:goto600
590 goto625
597 rem
598 rem set the heaters on or off
600 if t< dt(i+j) then oc=(oc and (255-2↑(i+j))): poke 56832, oc  : goto625
620 oc=(oc or (2↑(i+j))):poke 56832,oc
625 next i
630 j=j+4
635 next p
637 rem check the extra, if used do it
640 if e$="n" then 695
645 if x>0 then i=0:p=56842:ef=1:j=0:gosub 176
650 if y>0 then i=1:ef=1:  gosub 176
655 if z>0 then i=2:ef=1: gosub 176
660 if s>0 then i=3:ef=1: gosub 176
665 if m>0 then i=0:p=56843:ef=1:j=13:   gosub 176
670 if g>0 then i=1:ef=1: gosub 176
675 if q>0 then i=2:ef=1: gosub 176
680 if f>0 then i=3:ef=1: gosub 176
685 rem all operating chan's were used
690 ef=0: j=0
695 next l
700 rem ******************************
705 rem  set and change the display
710 rem ******************************
720 print "[illegible]";
730 for a=0 to 7  step 2
740 h(a)=int(h(a)/10):h(a+1)=int(h(a+1)/10)
750 print tab(50)h(a);" ";tab(30)h(a+1);" [illegible]";
```

```
755 next a
760 print "■■■■";
765 if e$="n" then 164
767 print "■";
770 for a=8 to 15 step 2
772 h(a)=int(h(a)/10):h(a+1)=int(h(a+1)/10)
775 print tab(50)h(a);" ";tab(30)h(a+1);" ■";
780 next a:
897 goto 167 :rem    loop again !
900 rem     ======================
902 rem      fix display format
903 rem     ======================
904 for a=1 to 50: print " ":next a
908 h$="current temperature readings"
909 ht$="heater":eq$="#":c$=","
910 ext$="extra operating channels"
915 print"■■■";tab(5)h$;tab(82)ht$;"1=";tab(22)ht$;"2=";
916 print  tab(87)eq$;"3=";
917 rem
920 print tab(27)eq$;"4=";
922 print tab(87)eq$;"5=";tab(27)eq$;"6=";
925 print tab(87)eq$;"7=";tab(27)eq$;"8=";
926 rem
927 print "■■";tab(88)ext$;"■";
930 print tab(87)eq$;"9=";tab(26)eq$;"10=";tab(86)eq$;"11=";tab(26)eq$;"12=";
932 rem
935 print tab(86)eq$;"13=";tab(26)eq$;"14=";tab(86)eq$;"15=";tab(26)eq$;"16=";
940 return  : rem back to action
950 rem exttra initi. part
954 gosub 1000
955 input"are you using ch # 9";wh$
956 if wh$="y" then x=1
960 input"are you using ch # 10 ";wh$
961 if wh$="y" then y=1
962 gosub 1000
965 input"are you using ch # 11";wh$
966 if wh$="y" then z=1
967 gosub 1000
970 input"are you using ch # 12";wh$
971 if wh$="y" then s=1
972 gosub 1000
975 input"are you using ch # 13";wh$
976 if wh$="y" then m=1
977 gosub 1000
980 input"are you using ch # 14";wh$
981 if wh$="y" then g=1
982 gosub 1000
985 input"are you using ch # 15";wh$
986 if wh$="y" then q=1
988 gosub 1000
990 input"are you using ch # 16";wh$
991 if wh$="y" then f=1
992 goto 161
1000 print "■"
1005 for a=1 to 44
1010 print "
1020 next a
1025 print "■■■■■■■■■■":return
1100 rem #### undershoot set routine ##
1200 for a=0 to 7
```

```
 1210 dt(a)=dt(a)-60
 1220 next a
 1230 ud=1 :rem set delay variable
 1240 goto 167 :rem goto main program
 1300 rem delay until expected desired temperature
 1310 poke 56832,255:rem heaters off
 1320 for a=0 to 7
 1330 dt(a)=dt(a)+60
 1340 nexta
 1350 dl=1: rem set delay flag
 1360 ud=0
 1370 goto 167 :rem goto main program
 1500 rem ****** timer sub-prog. ****
 1505 print "note time should be given in military notation"
 1510 input "what time is it now";ftime
 1515 gosub 1000
 1520 rem change the times to minutes**
 1525 ftime=100*((ftime/100)-int(ftime/100))+(int(ftime/100))*60
 1530 rem
 1535 input "what is the start time for operation";sime
 1540 rem
 1545 sime=100*((sime/100)-int(sime/100))+(int(sime/100)*60)
 1550 rem
 1555 gosub 1000
 1560 print "the system can be set for next days too"
 1565 print "give right number corresponding to the day you wish to operate"
 1570 print "'0' for today, '1' for next day and add one for each day"
 1575 input "now what day you want to operate";n
 1580 gosub 1000
 1585 print "[illegible]*** system is under pre-timed program***"
 1590 rem
 1595 rem delay until the final time
 1600 et=n*(1439)-ftime+sime
 1605 for i=1 to et
 1610 for a=1 to 39000: next a
 1615 next i
 1620 return :rem back to main prog.
ready.
```

HEATERS ON/OFF

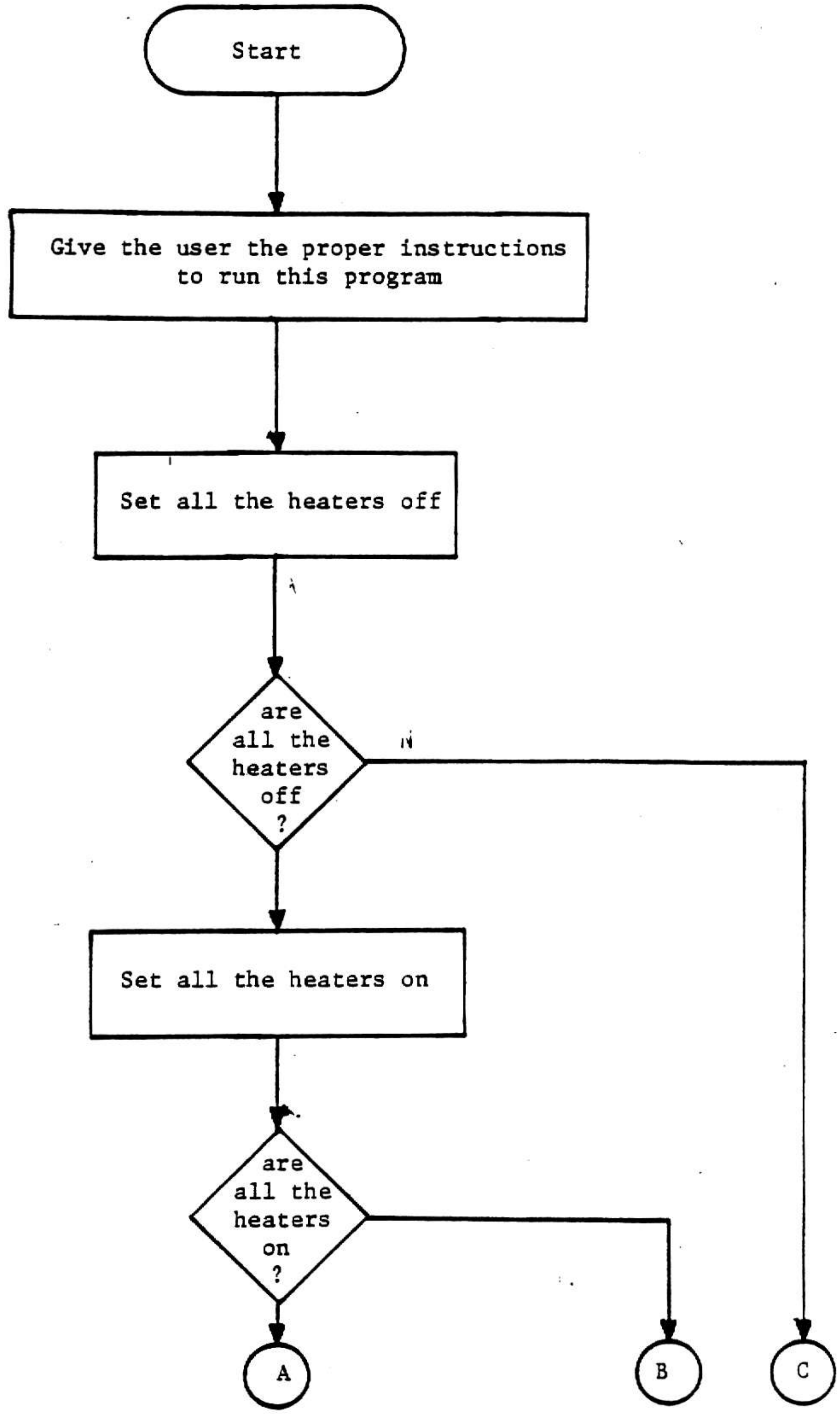


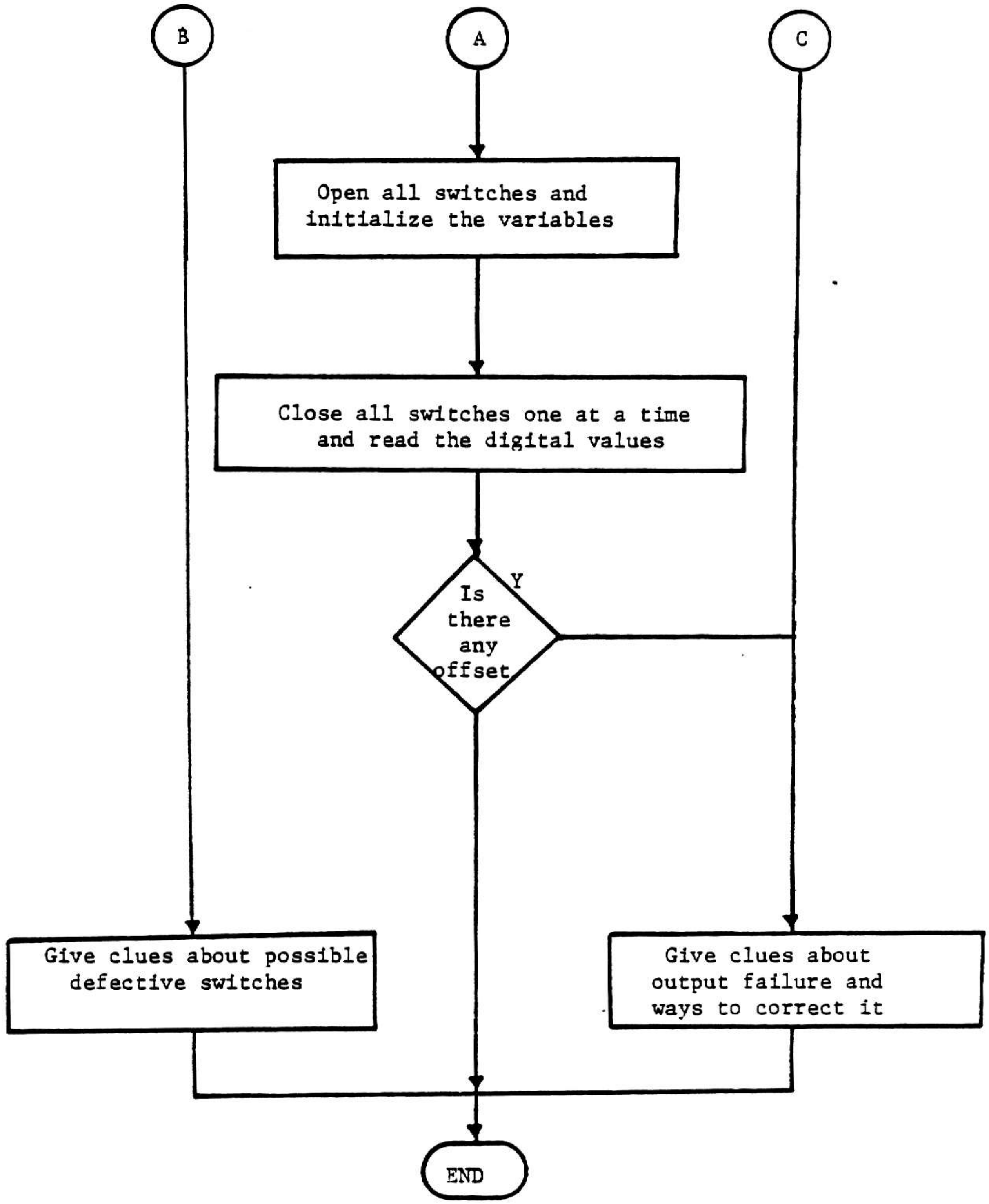
HEATERS ON/OFF FLOWCHART

```
ready.

 10 rem heaters on/off
 15 rem open all input analog switches ,
 17 rem leave s/h in sample state ,and
 20 rem turn off all heaters.
 24 rem h is the array heaters ; h(1)=1
 27 rem means heater one is on and h(1)=0 means heater one is off.
 30 dim h(8)
 35 rem
 40 for i=56840 to 56843
 45 poke i,0
 50 next i
 55 poke 56839,0:rem s/h in sample state
 60 rem clear the screen
 65 print"■"
 70 for i=1 to 40:print "                                        ":next i
 72 print "starting from heater # 1 do"
 75 print "initi. the heaters' array ;put "0" for the on heater and "1" for off"
 80 input h(1),h(2),h(3),h(4),h(5),h(6),h(7),h(8)
 85 oc=h(8)*2↑7+h(7)*2↑6+ h(6)*2↑5 + h(5)*2↑4+h(4)*2↑3+h(3)*2↑2+h(2)*2+h(1)
 90 rem
 95 poke 56832,oc
 100 print "heaters are set as desired"
 110 end
ready.
```

PERFORMANCE TESTS


Start
Give the user the proper instructions to run this program
Set all the heaters off
are all the heaters off ?
N
Set all the heaters on
are all the heaters on ?
A
B
C


B
A
C
Open all switches and initialize the variables
Close all switches one at a time and read the digital values
Is there any offset
Y
Give clues about possible defective switches
Give clues about output failure and ways to correct it
END

```
ready.

5 rem performance test
10 rem this program is written to facilitate the maintenance of mbhc
15 rem
20 rem with cooperation of the user this prog will give the list and clues
25 rem of the deffective chips
30 gosub 1000 :rem clear the screen
35 print"before proceeding further, please read and do the following"
36 for j=1 to  2000     :next j
40 gosub 1000
45 print "all the leads of the input channels should be connected to ground"
46 print" "
50 print"have the multimeter ready,and prepared at ac-1000"
51 print" "
55 print"if the above instructions are fulfilled press any key, else ";
60 print"stop this program and do them"
61 print "  "
65 input i$:rem dummy input
67 gosub 1000
70 print " test is in process"
71 for i= 1 to 2000 :next i
75 for i=56840 to 56844
80 poke i,0
90 nexti
95 poke 56839,0:rem sample and hold is in sample
100 poke 56832,255:rem set heaters off
105 gosub 1000
110 print"using the meter make sure that voltages across all the heaters are";
115 print " less then 10 volts"
120 print" "
125 print"are all the voltages less then ten,y/n"
126 input c$
130 if c$="n" then gosub 400
131 gosub 1000
135 poke 56832,0:rem heaters are on
140 print" "
145 print"using the meter make sure that voltages across all the heaters are";
150 print " greater than 215 volts"
155 print" "
160 print "are all the voltages greater than 215,y/n"
161 input g$
165 if g$="n" then gosub 400
170 gosub 1000
175 a=0
180 for p=56840 to 56843
185 for i=0 to 3
190 for k = 56840 to 56843
195 poke k,0:next k
200 for l= 1 to 10
205 poke p,(17*(2↑i))
210 for d=1 to 20: next d
215 poke 56836,0 :rem s/h in hold
220 poke 56837,0 :rem run a/d
225 if peek(56833)>=128 then 225
230 r=peek(56835)*16 + peek(56836)
233 poke 56839,0: rem s/h in sample
235 a=a +r
```

```
240 next l
245 if a/10 >50 then        500
250 next i
255 next p
265 gosub 1000
270 print" performance test is done"
275 end
400 rem  maintenance clues to output failure
405 gosub 1000
410 print" the source of failure might be the output latches, or the triacs"
415 print" "
420 print"check the control line to the corresponding optocoupler";
425 print" of the wrong reading"
430 print" "
435 print " if you want to proceed then press any digit key"
436 input t$
440 return
500 rem maintenance clues for i/p part
505 print"due to i/p analog failure please do the following instructions"
510 print" "
515 print "first turn the mbhc's power off "
520 print "take all the ana. switches  out "
530 print "test each ana. switch in laboratory for less than 5";
535 print " millivolts offset"
536 print" "
540 print "each switch should have min. of 3 millivolts offset "
545 print " "
550 print"replace the defective  switches and run this program again"
555 end
1000 print"[illegible]"
1010 for i= 1to 40
1015 print"                                                  "
1020 next i
1025 print"[illegible]"
1030 return
ready.
```

MICROCOMPUTER BASED HEAT CONTROLLER

by

MAZEN AKKAM

B. S., Kansas State University, 1983

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

ABSTRACT

The design of digital control system for chemical reactors operated by the Chemical Engineering Department of Kansas State University is presented. The control system keeps the temperatures within the reactors without substantial deviation from a set point. A Microcomputer Based Heat Controller, MBHC, was built using a Commodore 64 Computer in conjunction with integrated circuits such as a 12 bit Analog to Digital Convertor, a Sample and Hold, Analog switches, CMOS Patches, and Decoders.

Four programs work interactively with the user. The "Control Program", which implements feed forward and feed back control, carries out the control process by reading voltages of thermocouples, then signaling to an A/D to start converting the amplified voltage held by a sample and hold. The temperature corresponding to the voltage of the thermocouple is found and compared to a desired temperature provided by the user. Heaters within the reactor are then turned on or off.

The circuit was able to control the temperature within three to four degrees centigrade.

1430-67
CD-53