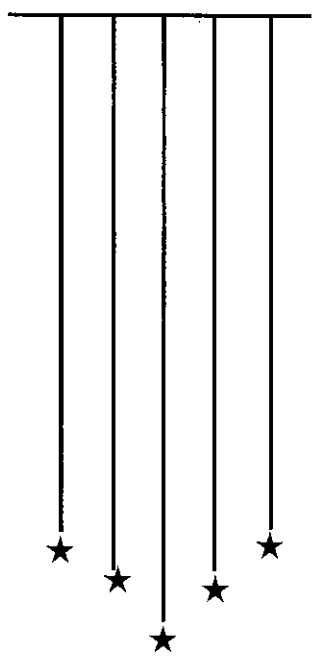
MICROPROCESSOR BASED PROGRAMMABLE LOGIC CONTROLLER

PROJECT REPORT



2000 - 2001

SUBMITTED BY

S. NITHYAPRIYA

G. MAHADEVI

GUIDED BY

Mrs. M. SHANTHI, M.S.,
Senior Lecturer, ECE Dept.,

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

ELECTRONICS & COMMUNICATION ENGINEERING

OF BHARATHIAR UNIVERSITY

Department of Electronics & Communication Engineering

Kumaraguru College of Technology

Coimbatore - 641 006.

Kumaraguru College Of Technology

Coimbatore - 641 006

Department of Electronics and Communication Engineering

Certificate

This is to certify that this project entitled

Microprocessor Based Programmable Logic Controller

Has been submitted by

Ms/Mr. .....

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineeering in the Electronics and Communication Engineering Branch of Bharathiar University, Coimbatore - 641 046 during the Academic year 2000-01

..... (Guide)

..... (Head of the Department)

Certified that the candidate was examined by us in the Project Work.

Viva-Voce Examination held on .....

University Register Number

.....

..... (Internal Examiner)

..... (External Examiner)

PREMIER INSTRUMENTS & CONTROLS LIMITED

pricol

PLANT I
P.B. No 4209, PERIANAICKENPALAYAM
COIMBATORE - 641 020, INDIA
PHONE : ++91 - 422 - 892901 TO
FAX : ++91 - 422 - 892026
e-mail : factory@pricol.jet.co.in
pricol.fact@sm8 sprintrpg.ems.vsnl.net.in

HRD/PROJ-TRG/2001
08-March-2001

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the following students doing final year BE (ECE) at Kumaraguru College of Technology – Coimbatore, underwent project work in our organisation.

⇒ *G Mahadevi*
⇒ *S Nithyapriya*

The details are :

- Title of Project : *MICROPROCESSOR – BASED PROGRAMMABLE LOGIC CONTROLLER*
- Period of Project : *November 2000 to March 2001*
- Department : *Production & Supply (Electronics)*
- Attendance & Conduct : *Good*

We wish them the very best for a bright future.

ANTHONY THIAGARAJAN
DEPUTY MANAGER – HUMAN RESOURCES

REGD OFFICE : P.B. No. 6331, 1087A AVANASHI ROAD, COIMBATORE - 641 037, INDIA
PHONE : ++91 - 422 - 211520 TO 4 FAX : ++91 - 422 - 210022
e-mail : factory@pricol.jet.co.in • pricol.fact@sm8.sprintrpg.ems.vsnl.net.in

BVQI ISO 9001

Acknowledgement

We wish to express our regards and sincere thanks to our principal Dr.K.K.PADMANABHAN,B.Sc.,(Engg),Tech,Ph.D., for providing us facilities to carryout this project.

We feel highly dedicated in manifesting our deep sense of thankfulness to our HOD Prof.RAMASWAMY M.E.,MISTE, MIEEE (USA) for his motivation and suggestions during the course of this project.

Our special thanks to our Internal project guide, Mrs.SHANTHI, MS., Senior Lecturer of electronics and communication department for her immense help in accomplishing this work.

We are greatly indebted to this estimable organisation PRICOL for extending their helping hand in completion of our project.

We would like to express our sincere Gratitude and thanks to ANTHONY THIAGARAJAN Dy.MANAGER from the HRD who was very helpful in granting us permission to carry out our project in PRICOL.

We also express our heartfelt thanks to Mr.V.MAHESH.ME.,Manager, P & S 5 Unit, whose help can never be comprehended. His efficient encouraging and enormous help made us see through in our project.

We also own our thanks to our classmates, teaching and non-teaching staff of Electronics and Communication Department.

Synopsis

The design and development of a Microprocessor based programmable logic controller is discussed in this project report. We have also included in our project the study of replacement of the electronic control of the gear hobbing machine by PLC

This system can handle a maximum of 8 inputs and 8 outputs and has a memory capacity of 512 bytes.

An external keyboard is part of the system which facilities the entry of control program by using 12 instruction keys 16 hexadecimal keys, 3 special command keys and one execution key. The software method of key scanning is employed. The system also has an I/O simulator to simulate the input conditions for any control application of interest. A set of LED'S are used to indicate the input as well as output status.

Regarding the implementation of PLC technique in gear hobbing machine, previously it was working under the principle of TTL logic by making use of eight PCB's Each PCB has the control of performing a particular set of functions of the gear hobbing machine. This requires more space and components and would also increase the cost of the system. Thus for large machines the relay panels are too complicated. In such large machines the PCBS and numerous relays can be replaced by a programmable logic controller.

The PLC is operated by a program which is encoded by a programmer. Hence trouble shooting becomes much easier in case of a PLC. All the functions can be controlled by a switch operation and it can be altered according to the data in the program.

CONTENTS

CHAPTER - 1

INTRODUCTION

Many of the automation operations involve basically sequencing interlocking and logic control. These controls involving predominantly binary type inputs and outputs have been traditionally realised through manual control, pneumatic control, electromagnetic relays, or hardwired electronic i.e. logic. The most common implementations were the last two. Among these the relay approach is bulky, expensive and slow, has shorter life of operation and doesn't lend itself easily when modifying the systems configurations. On the other hand, the so-called hardwired system, the logic is to be entirely redesigned and wired afresh due to change in the pattern of the production.

A new approach in this area is the programmable logic controller (PLC) Besides overcoming the short coming of relay logic and hardwired systems. The PLC is designed with facility to alter the control sequence by simple reprogramming without major changes in the hardwire. Making it superior to relays and conventional hardwired system.

The objective set forth for this project work is to design and develop a microprocessor based PLC catering 8 Inputs and 8 Outputs. It is designed to handle a set of 12 instructions.

The basic objective in replacing the electronic control of gear hobbing machine by PLC is to produce uniformly with in scheduled time so as to increase production. We have also presented the main drives, main parts and the working sequence of gear hobbing machine, along with the realization of logic functions and the main program on which the gear hobbing machine works.

CHAPTER - 2

PLC HARDWARE

2.1. Definition:

The programmable logic controller has been defined by National Electrical Manufacturers Association (NEMA) of USA means the following "A digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions for implementing specific functions such as logic, sequencing, Timing, Counting and Arithmetic to control through digital or analog input/output unit, various types of machines or processes". A digital computer which could be used to perform the functions of a programmable logic controllers is considered to be within this scope.

2.2. PLC Basics:

Regardless of size, cost of complexity, all PLC'S share the same basic parts and functional characteristics. As shown in the block diagram of figure (1), a PLC typically consists of the following.

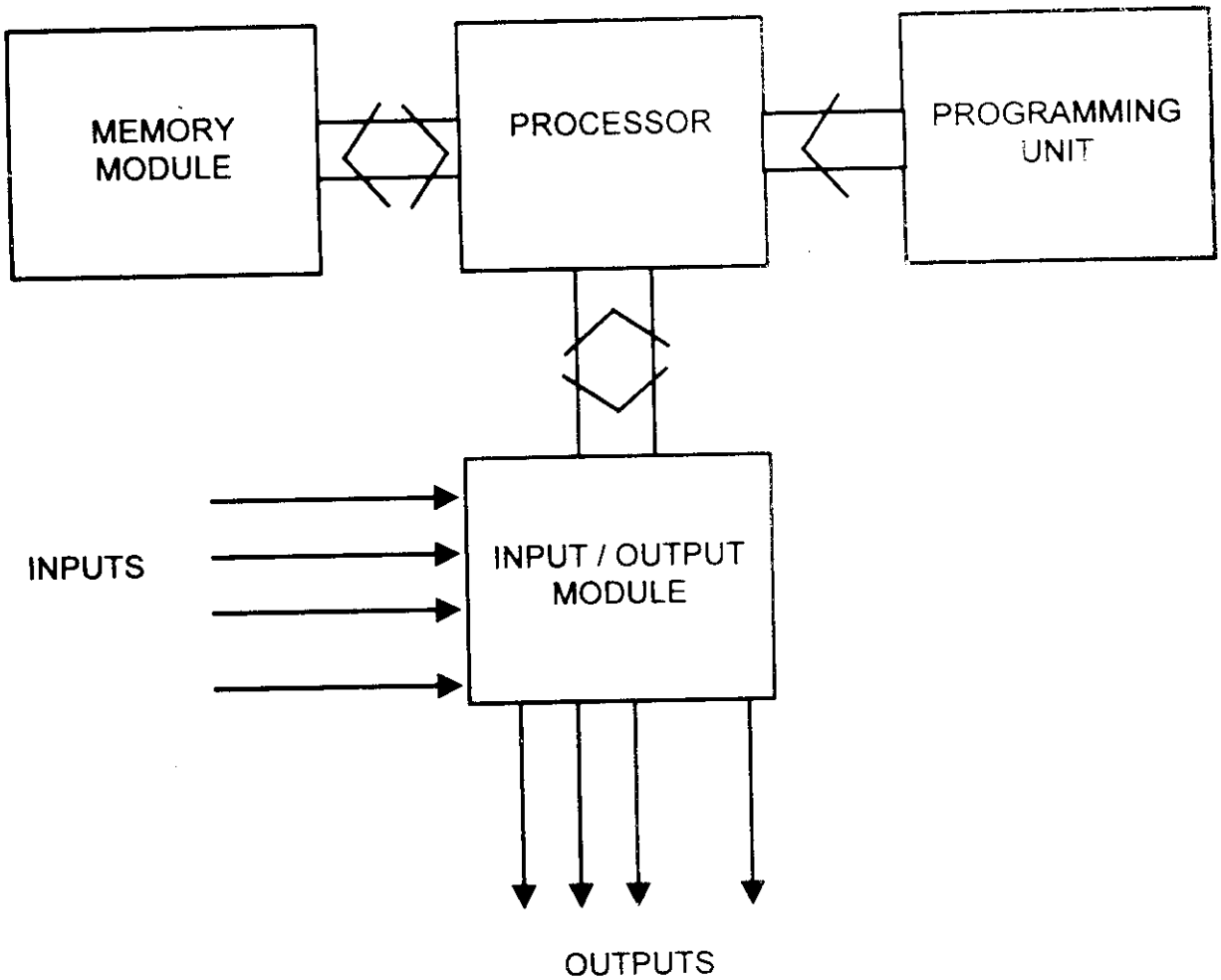
a. Power Supply Unit

b. Processor Unit

c. Memory Unit

d. Programming Unit

e. Input and Output Interface Unit.



BLOCK DIAGRAM OF THE PROGRAMMABLE CONTROLLER

Figure -1

Functionally, a PLC examines the status of input interfaces and in response, controls something through output interfaces. Combinations of input and output data are referred to as logic. Several logic combinations are usually needed to carry out a control program. This control program is stored in memory using programming unit. All logic combinations stored in memory are periodically evaluated by the processor in a predetermined order. The period required to evaluate the logic combinations is called a scan.

2.2.1. Power Supply Unit:

Most PLC'S operate on +5V DC and - 5V DC. So we use +5V DC Regulator power supply. This power supply unit comprising a230V/ 10V-AC - 500 m amps step down transformer associated with 4 x IN4007 diode in bridge configuration by using electrode filtering capacitor, Regulator Chip (7805) respectively.

2.2.2. Processor Unit:

The microprocessor, sometimes called the central processing unit is the heart of the PLC and organises all control activity. The CPU electronically scans the control program stored in memory along with the status of inputs and executes a specified command to the appropriate output. A clock for timing the entire system is part of the processor unit. In addition to straight logic processing, the processor may perform other functions such as timing, counting, decision making and arithmatics.

2.2.3. Memory Unit:

This unit stores the control program for the machine or process being controlled. This control program or information stored in the memory relates to how the input and output data should be processed. The amount of memory required depends on the complexity of the control program.

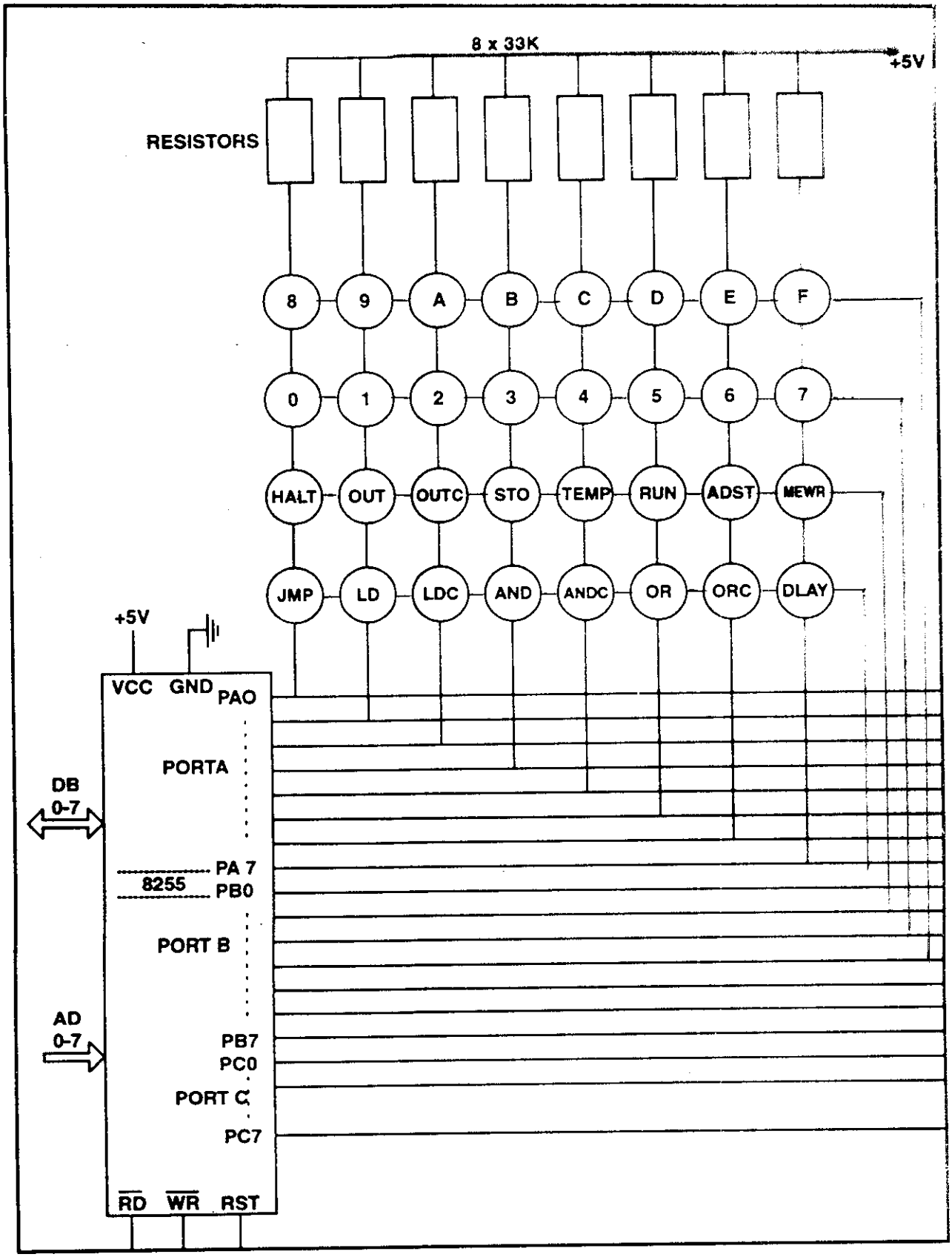
2.2.4. Programming Unit:

The programming unit is a 4 x 8 matrix keyboard as shown in figure(2). It has one Run key, 12 instruction keys, 16 hexadecimal keys, 3 special command keys named as TEMP, MEWR and ADST. The 'TEMP' key is for temporary storage of data. The 'MEWR' key is a memory write and 'ADST' key is a address set. The control program are written in terms of 12 instructions and 3 special commands.

2.2.4.a. Instruction Set:

The PLC systems has a set of 12 instruction which are normally found in most of the commercial PLC'S. The instruction set is listed below.

Mnemonic	**Action:**
LD	Load Input Status
LDC	Load Input Status Complement
AND	Logically AND Input Status
ANDC	Logically AND Input status complement
OR	Logically OR Input status
ORC	Logically OR Input status complement
DLAY	Generate Delay
HALT	Halt
OUT	Output Logic Result
OUTC	Output logic result complement
STD	Store Input / Logic Status
JMP	Jump to Start



ARRANGEMENT OF KEYS WITH SOFTWARE SCANNING

Fig.2

2.2.4.b. Key Sequence:

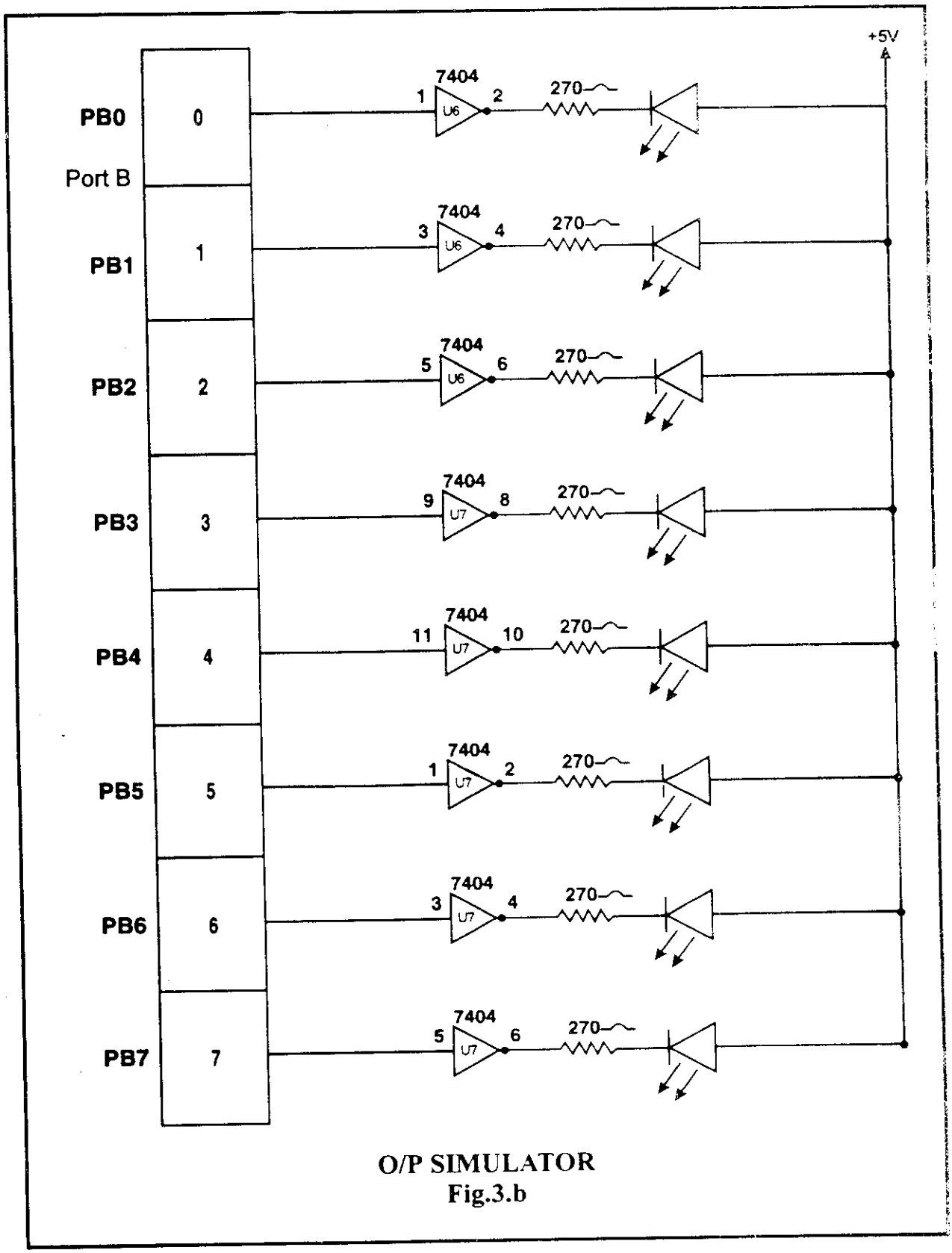
The inputs are handled by the following sequence of keystrokes. The keys LD, LDC, AND, ANDC, OR, ORC are pressed followed by any one of the digit keys O through 7. In order to effect any delay in the output side, the key 'DLAY' Must be followed by any of the digit keys O through F, the keys OUT and OUTC must be followed by any one of the digit keys 8 through F. The keys 'JMP' and 'HALT' must be followed by the 'O' digit key. The key 'STO' must be followed by the 'TEMP' key. On violation of the above key sequence the system displays an error message when the conversion program is executed.

2.2.5. Input and Output Interface Unit:

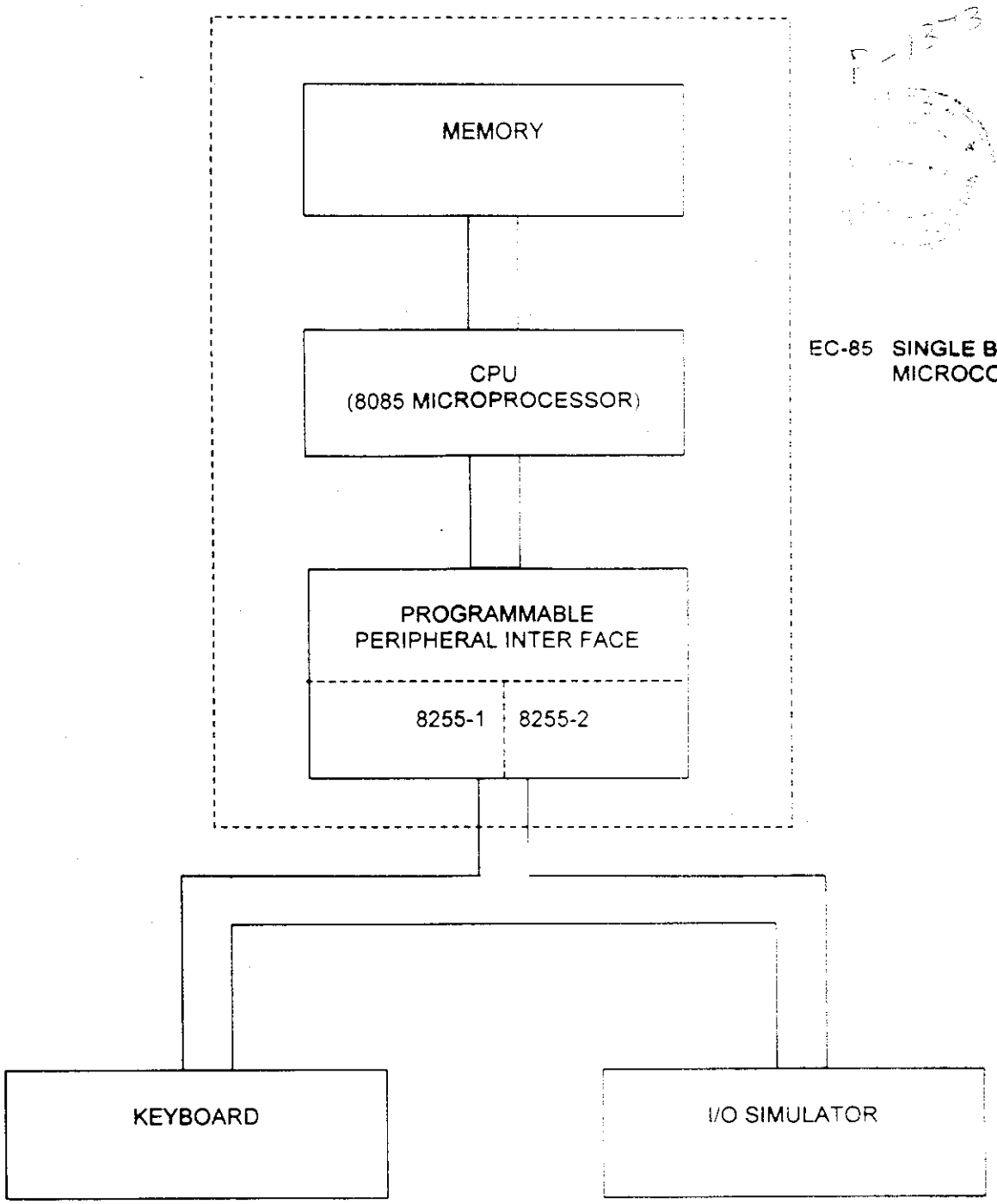
Input unit accept signals from the machine or process elements and translate them into signals which the PLC can accept. The input signals may be from limit switches, push buttons, Pressure switches, other on-off devices or transducers.

Output unit convert controller signals into external signals required for controlling the machine or process. These outputs may be devices such as solenoids, starters, relays, positioning valves etc. In other words, changing low DC logic signals from processor to high DC or AC output signals required for driving the loads is the function of output unit.

In this PLC, it has 8 inputs and 8 outputs. To aid in the simulation of the Input-output conditions, an Input-Output simulator has been configured as shown in Figures (3). A set of toggle switches are used for simulating the Inputs and the Output conditions are monitored by a set of LED'S. All the Inputs and outupts are interfaced to the microprocessor system through programmable peripheral Interface support chip (8255IC).



O/P SIMULATOR
Fig.3.b



EC-85 **SINGLE BOARD MICROCOMPUTER**

SYSTEM BLOCK DIAGRAM

Figure - 1

CHIP	ADDRESS	PORT - A	PORT B	PROT C
8255-1	03	INPUT (FROM KEYBOARD)	OUTPUT (TO KEYBOARD)	OUTPUT (NOT USED
8255-2	0B	INPUT (FROM I/O SIMULATOR OR INPUT UNIT)	OUTPUT (TO I/O SIMULATOR OR OUTPUT UNIT)	OUTPUT (NOT USED)

PORT ALLOCATIONS OF PPI CHIPS

Table - 1

INSTRUCTION KEYS	KEY CODES
JMP	10
LD	11
LDC	12
AND	13
ANDC	14
OR	15
ORC	16
DLAY	17
HALT	18
OUT	19
OUTC	1A
STO	1B
TEMP	1C
COMMAND KEYS	KEYCODES
RUN	ID
ADST	IE
MEWR	IF

DIGIT KEYS	KEY CODES
0	00
1	01
2	01
4	04
5	05
6	06
7	07
8	08
9	09
A	0A
B	0B
C	0C
D	0D
E	0E
F	0F

KEY CODES

Table - 2

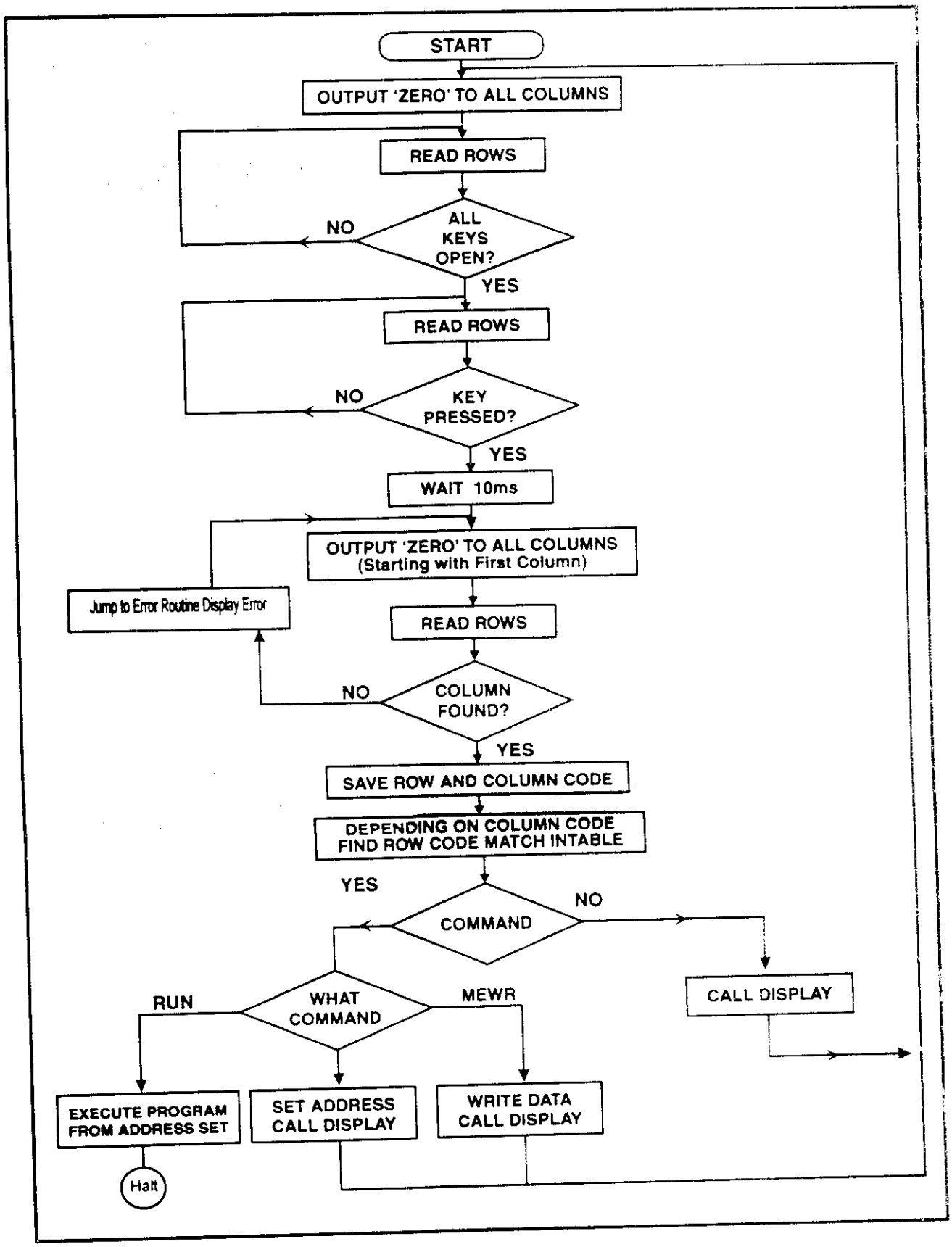
3.3.1. Key Scanning:

The PLC system employs the software scanning method of key scanning. Any key pressed is sensed by a key scan routine, and for each key its equivalent PLC code is stored in mernory location starting from location 2000H the flowchart for the key scan routine is shown in figure (3).

3.3.2. Keycodes and Keycode Conversion:

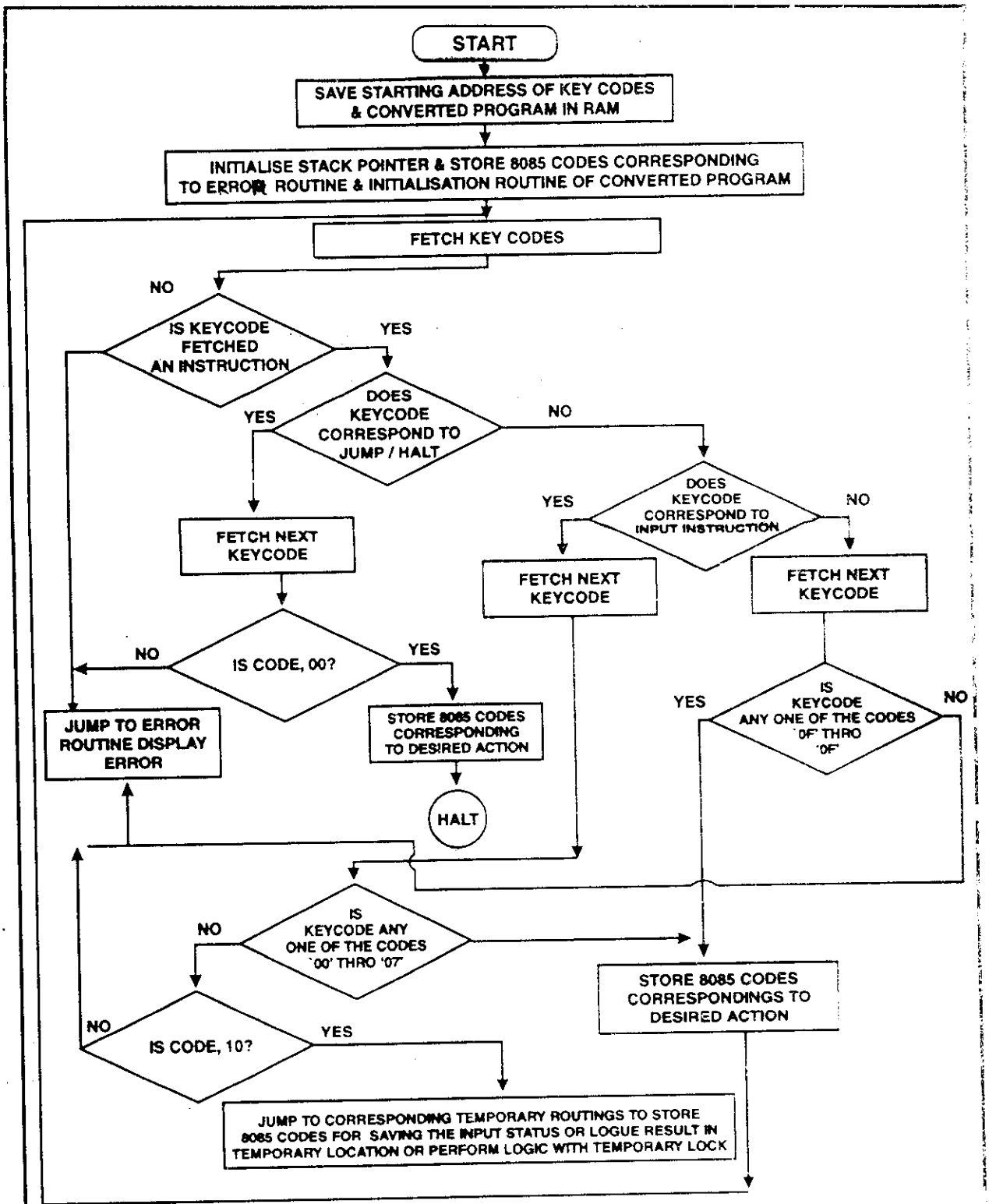
The key pressed is sensed by the key scan routines and depending up on the key pressed, the equivalent PLC code is stored in successive memory locations starting from 2000H. The keycodes are given in table (2).

The key codes are converted in to the equivalent 8085 machine code instruction by a program called "Conversion Program". The flow chart for conversion program is shown in figure (4).



FLOWCHART FOR KEY SCAN ROUTINE

Fig.3



CONVERSION PROGRAM FLOWCHART

Fig.4

3.4. System Working:

Using the original keyboard on the microprocessor kit (EC-85) the key scan routine starting from address 0800H it is invoked to accept the keycodes via the external keyboards. These keycodes get stored in consecutive location starting from 2000H. Thereafter the conversion program residing in location beginning with 0940H is executed using external keyboard. The converted program starting from location 2200H. To execute the program given by the converted codes the processor is set to the address 2200H and 'RUN' key is pressed. The required initiallisation is carried out by an initiallisation program a residing in the location 2200H through 220CH. The system is reset using the RESET key on the microprocessor kit. The key scan program is run again. The converted program is then executed using the external keyboard.

Once the converted program is executed the clock which is part of the processor stops the PLC sequentially so that each address of the memory is interrogated in succession. The inputs selected are logically processed in accordance with the instructions and depending on the result the appropriate output is energized or de-energized (indicated by the ON/OFF state of the corresponding LED).

3.5. System Software:

The software listing which includes keyscan rontine (monitor program) and conversion program are given in Appendix.B.

Memory Map:

The complete memory map of the system is shown in Table - 3 .

0000 - 07FF	EC-85 SYSTEM MONITOR (EPROM)
0800 - 093F	KEY SCAN ROUTINE (EPROM)
0940 - 0C5D	CONVERSION PROGRAM (EPROM)
2000 - 21FF	KEY CODES (PLC INSTRUCTION) (RAM)
2200 - 23FF	CONVERTED 8085 MACHINE CODES (RAM)

MEMORY OF THE SYSTEM

Table - 3

CHAPTER - 4

PLC SOFTWARE

4.1. Introduction To Programming The PLC:

There is a constant need for process control systems in the manufacturing industries to produce a better quality product more efficiently and at a low cost. To ensure that the plant or machinery operates with in the tolerances and at correct speed etc, it must be programmed and controlled.

With the development of computer technology for industrial applications, the programmable logic controller was developed to replace the discrete relay, timer and counter. The programming languages encountered here are as follows.

(i) Relay Symbology

(ii) Ladder diagram technique

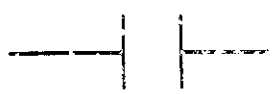
(iii) Boolean logic

(iv) Computer Language.

4.2. Details About PLC Software:

(i) Relay Symbology:

Control engineers, electricians and maintenance technicians have been using relays for many years. This Relay Symbology primarily provides the stability to examine the ON/OFF status of a specific bit addressed in memory and controls the state of an internal or external output bit. There are two types of relay instructions:

a. **Normally Open Contact:** 

The normally open contact is programmed when the presence of the input signal is needed to turn an output ON. When evaluated, the referenced address is examined for an ON(1), when examined, the referenced address is ON, then the

normally open contact will close and allow logic continuity (powerflow). If it is OFF(O), the normally open contact will assume its normal programmed state (Open), then braking logic continuity.

b. Normally Closed Contact: 

Normally closed contact is programmed when the absence of the reference signal is needed to turn on output ON(1). When evaluated the referenced address in examined for on off condition. When examined the reference address is OFF, then the normally closed contact will remain.

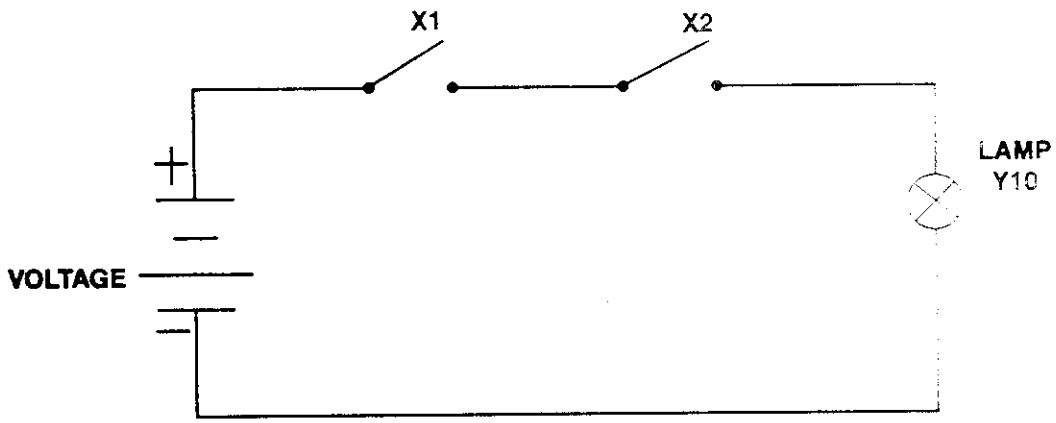
The examples of relay symbology circuit are shown in figure (1). The example (1) is a simple AND logic relay symbolgy circuit. In this example (1), the output is only obtained when both the swiftness applied to the system are loosed ie. they are both at logic 1.

The example (2) consist of a normally open start pushbutton in series with a normally closed push button. A holding contact of the program start relay is connected in parallel a cress the start push button. The operation of the start push button will allow power to the relay, thus closing the relay contact. When the start push button is released, the circuit will remain in the ON condition until the stop push button is operated to break the power to the relay.

ii. Ladder Diagram Technique:

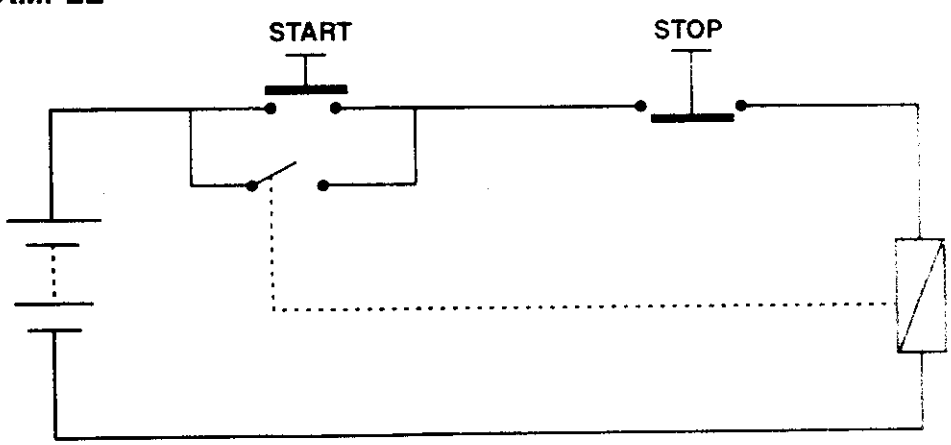
The ladder diagram technique is a symbolic instruction set which is used to crease a programmable control program. It is composed of six categories of symbolic instruction. They are relay type (open and closed contact), timer symbol, counter symbol, data manipulation, arithmetic, data transfer and program control.

1. EXAMPLE



(I) SIMPLE AND LOGIC CIRCUIT

2. EXAMPLE



(II) START / STOP LATCHING RELAY SYMBOLOGY CIRCUIT

RELAY SYMBOLOGY CIRCUIT

Fig.1

The ladder instruction symbols can be formatted to obtain the desired control logic that is to be entered in the desired memory. The ladder diagram emphasis the output which is dependent on the input condition. This control is accomplished through the use of a ladder rung. Which consist of a set of Input switches and output relay coil. Coils and contacts are the basic symbol of the ladder instruction sets. All inputs are represented by open and closed contats. All outputs are represented by coil symbol.

The examples of ladder diagram technique are shown in figure (2). The two examples are only using open contact, Closed contact and relay coil. This examples shows that ladder diagram is reduced version of the relay circuit.

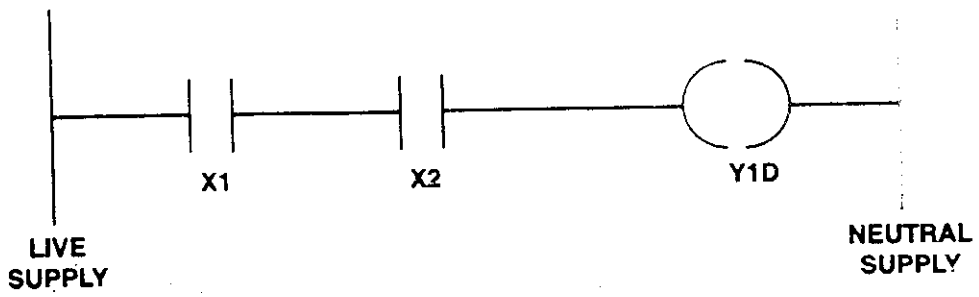
iii. Boolean Logic:

Any ladder diagram depends up on Boolean logic like AND, OR, EXOR, NOT, NAND, NOR etc. This Boolean logic are translated into computer languages. In previous examples, First one is simple AND Logic circuit and second one is combination of AND-OR Logic circuit. The various type of Boolean logic circuits are shown in figure(3).

iv. Computer Language:

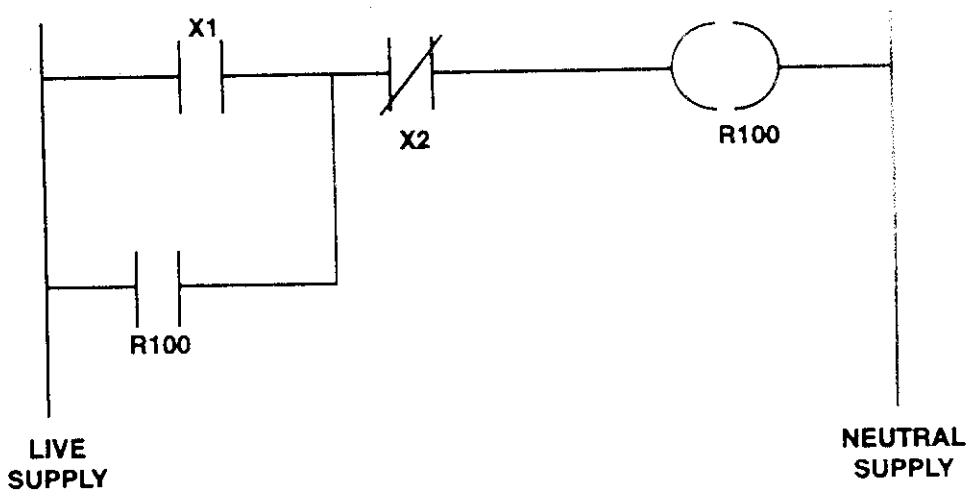
Finally, the ladder diagram is written into program by any computer language. Our project uses assembly level language. Assembly level language program is then loaded in to memory location and this will be processed by Microprocessor. Commercially, high level language is used in PC based control program.

1. EXAMPLE



(I) SIMPLE AND LOGIC LADDER DIAGRAM

2. EXAMPLE



START / STOP LATCHING - LADDER DIAGRAM

LADDER DIAGRAM TECHNIQUE

Fig.2

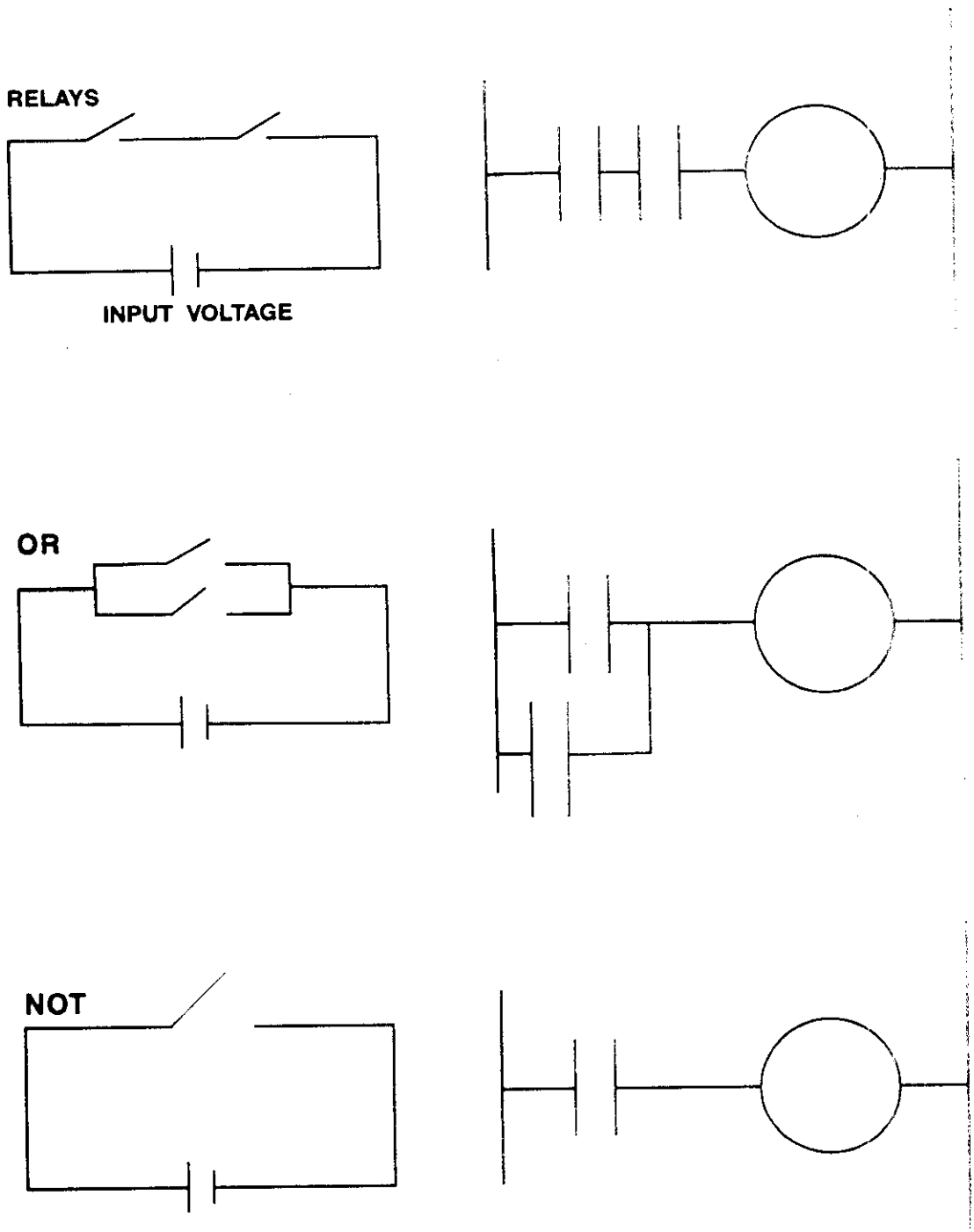
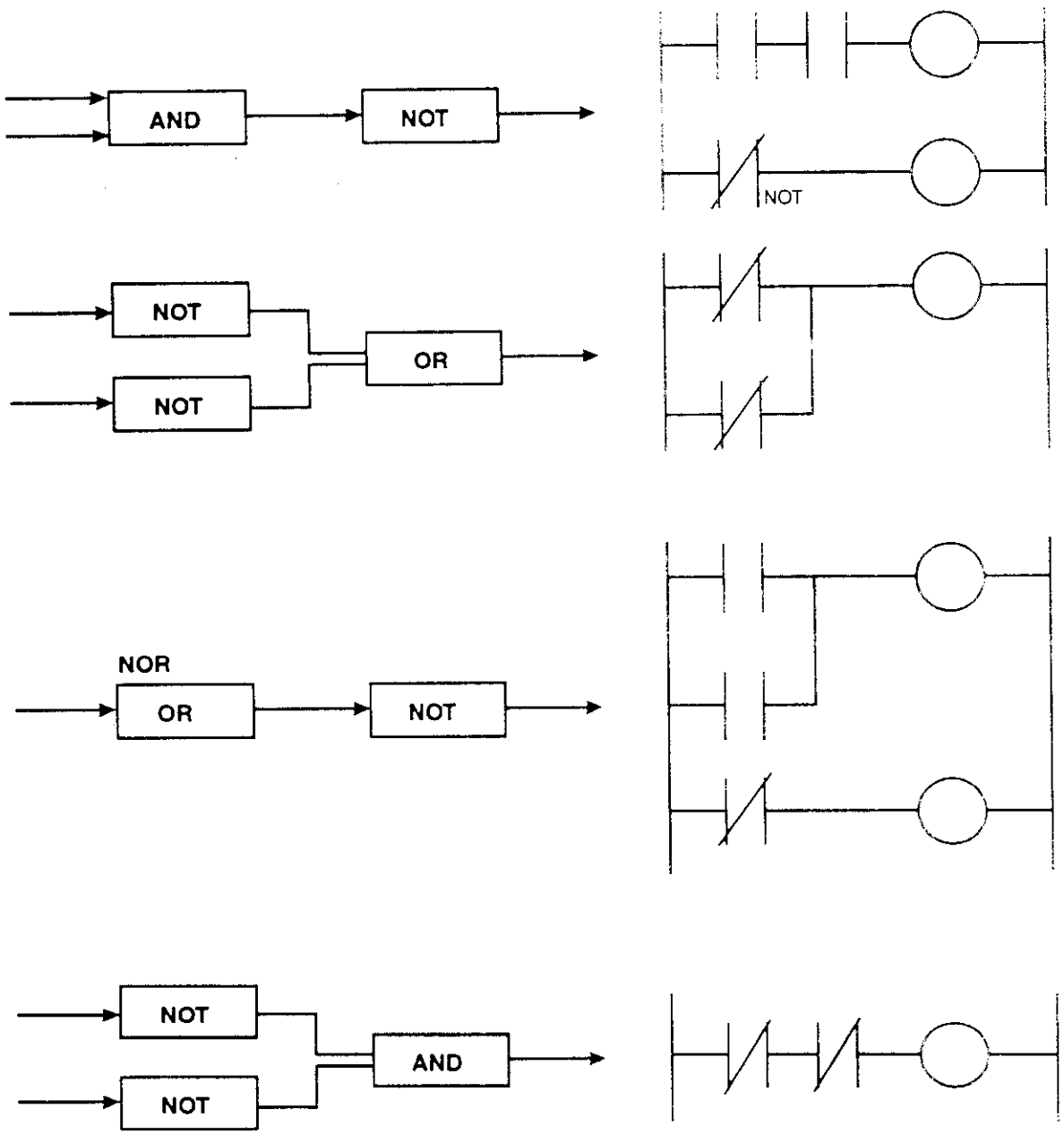


FIG. 3.a REALISATION OF LOGIC FUNCTION IN LADDER DIAGRAMS



NAND CAN BE REALISED IN TWO WAYS

Fig.3-b

4.3. Process Scanning Considerations:

PLC's functions by scanning their operational programs. Process scanning method or format is shown in figure 4. Each PLC operational cycle is made up of three separate parts:

- Input Scan
- Program Scan
- Output Scan

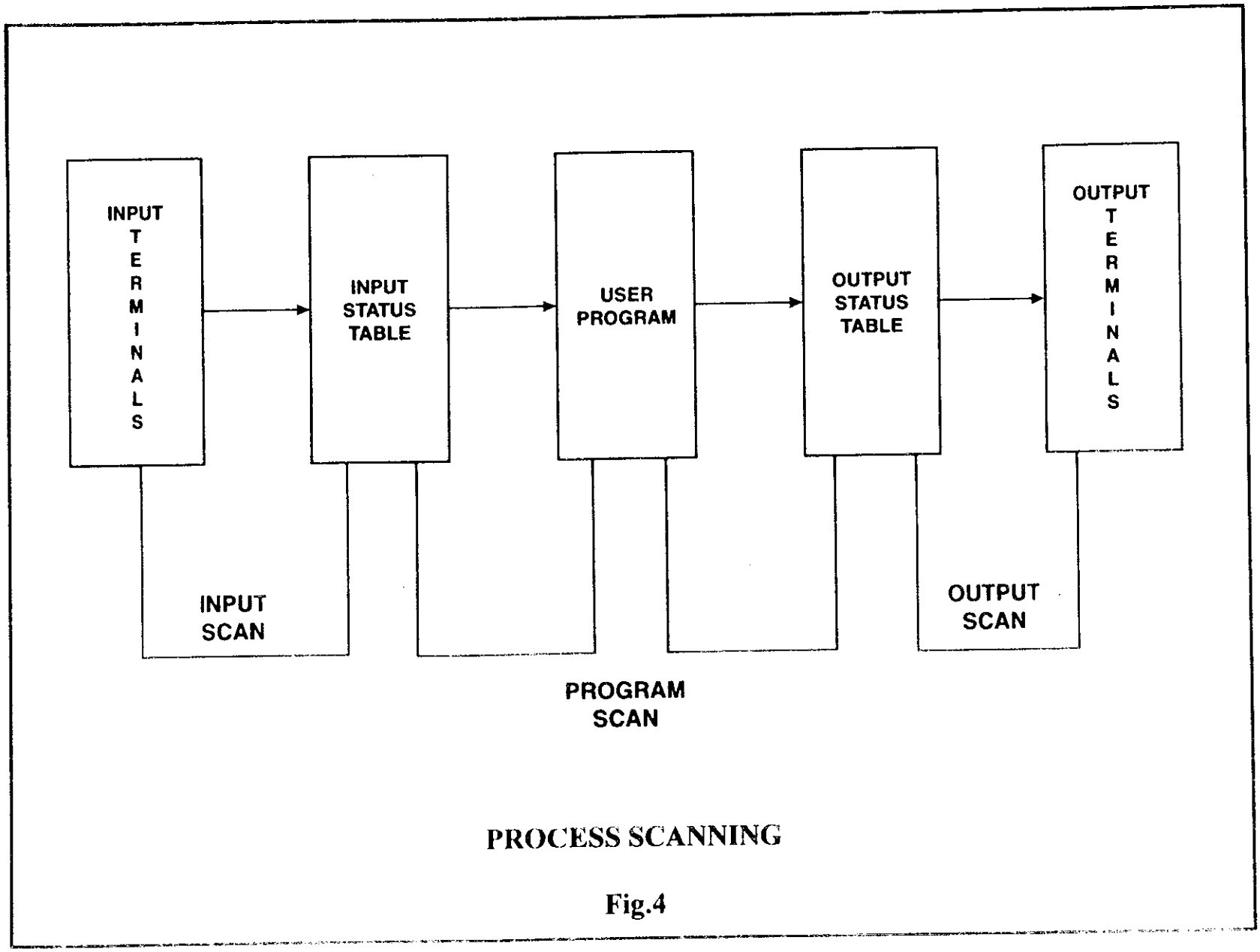
The total time for complete program scan is a function of processor speed and length of user program.

The PLC initially reads the ON/OFF condition of the inputs used in the program. These conditions are then stored in the input image memory during input scan.

During the program scan, data in the input status table is applied to the user program. The program is executed (Instructions carried out in the sequence) and the output status table is updated appropriately.

During the output scan, data associated with the output status table is transferred to the output terminals.

The input, program and output scans are separate independent functions. Typically the complete ladder scan time is a few milliseconds.



PROCESS SCANNING

Fig.4

To be able to design a PLC program using a computer, it is essential for the PLC software to have the following facilities:

- Programs can be designed using conventional ladder diagram technique.
- Test the program whether it is valid for use on the chosen PLC.
- Programs can be permanently saved either on the computer's hard disk or on floppy 3.5-inch disc.
- Programs can be re-loaded from either the hard disc or the floppy disc.
- Ladder diagram contacts and coils can be annotated with suitable comments.
- Hard copy printouts can be obtained.
- The programs can be transferred to the PLC via a serial link.
- The program with the PLC can be transferred back to the computer.
- Ladder diagram controls can be monitored in real time.
- Modifications can take place while the PLC is on-line.

The programs can be constructed either ON-LINE or OFF-LINE as required.

On-Line Programming:

It is implemented with direct connection to the PLC where changes to the program may be made while the PLC is in running mode. Timers, counters and other instruction formats can have their contents altered.

Off-Line Programming:

OFF-LINE programming is used to construct new programs prior to running process. This allows any programs to be checked and verified by the user and avoids the problems of any incorrect commands to the process i.e., invalid instructions and/or input/output references. This is the preferred method of programming a PLC. This type of programming format generally used for PLC's is Ladder Logic. This is the graphical representation of the hard-wired systems used

by electricians and includes switches, timers, counters and many other instructions in a structured way.

4.4. Proper Construction Of PLC Ladder Diagrams:

A PLC programming format's limitations must be observed when programming a PLC ladder diagram otherwise, the PLC's CPU will not accept the screen-programmed ladder diagram in to it's memory. In some cases, when incorrectly formatted ladder diagrams are received, an error message appears on the screen.

Examples of the limitations of a typical PLC are:

- A coil must be inserted at the end of rung.
- All contacts must run horizontally. No vertically oriented contacts are allowed.
- The number of contacts per matrix is limited.
- Only one output may be connected to a group of contacts.
- Flow must be from left to right.
- Contact progression must be straight across.

4.5. Latest Expansion In PLC Software:

In project PLC timer and counter instructions are not used. These instruction are latest version and this will improve PLC operation. The details about this instruction are given as below:

4.5.1. PLC Timers:

The most commonly used process control device after coils and contacts is the timer. Typical industrial timing tasks include timing of intervals for welding, painting and heat treating. Timers can also pre-determine the interval between two operations.

The most common timing section is TIME-DELAY ON, which is the basic function and there are also many other configuration which can be derived from one

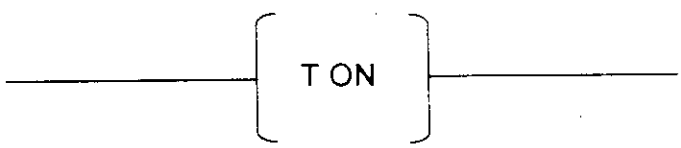
or more of the basic TIME-DELAY ON functions. PLC's have the one basic timer capability in multiplier. Typical of the derived functions are TIME-DELAY OFF interval pulse timing of more than one process of operation.

Timer Instructions:

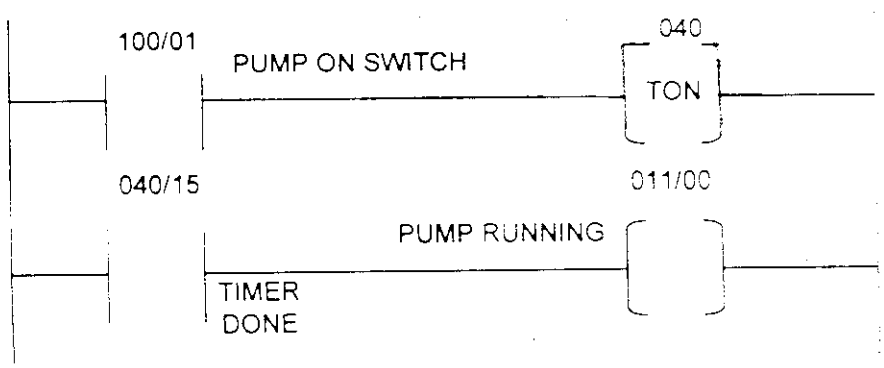
Timers are output instructions that provide the functions, as would hard ware timers. They are used to activate or de-activate a device after an expired interval. A timer counts the number of times that a fixed interval elapses. The timer instructions require an accumulator register to store the elapse count and a preset register to store a preset value. The preset value will determine the number of time-based intervals that are to be counted. When the accumulated value equals the preset value, a status bit is set on and can be used to turn on an output device.

There are basically two types of timer instructions.

ON-DELAY Timer Instruction:



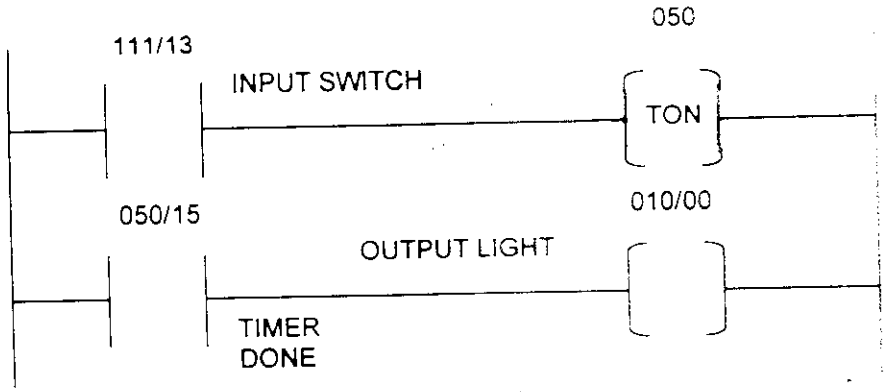
The ON-DELAY timer output instruction is programmed to provide time delayed action or to measure the duration of which some event is occurring. If any path has logic continuity, the timer begins counting time-based intervals and counts until the accumulated time equals the preset value as long as the rung conditions remain TRUE. When the accumulated time equals the preset time, a timer DONE bit is set to 1.Whenever the rung logic conditions go FALSE the accumulated value is reset to all zeros.



In the example application shown above, when the pump start switch is ON, the timer begins to count time-based intervals. As long as the switch remain closed or ON, the timer increments it's accumulated value for each counted interval. When the accumulated value equals the programmed preset value of 5 seconds, the timer stops incrementing it's accumulated value and sets the timer DONE bit 040/15. This DONE is then used in rung 2 to energise the pump running output bit 011/00.

OFF-DELAY Timer Instruction:

The OFF-DELAY timer instruction is programmed to provide time delayed action. If logic continuity is lost, the timer begins counting time-based intervals until the accumulated time equals the programmed preset value. When the accumulated time equals the preset time, the output is de-energised and the time bit is set to zero. the timed contact can be used through out the program as a NO or NC contact. if logic continuity is gained before the timer is timed out, the accumulated value is set to zero.



In the above ladder, if the input switch is open (logic) the timer will coat upto 5. When the present value is equal to the accumulated value, the time bit (050/15) will be set to zero and the output light will turn off in the second rung.

Advantages:

PLC timer may be a programmable variable time as well as at fixed time. Another advantage of the PLC timer is that it's time accuracy and repeatability are extremely high because it is based on solid-state technology. The PLC timing function is more versatile and flexible than the industrial or the digital electronic timers. In many control tasks there is a need to control time. To incorporate such control, PLC's are provided with build-in timers. Timers count fraction of second or seconds using internal CPU block.

4.5.2. PLC Counters:

Counters are used for simple operation as counting the number of paths made by the machine. They are also used for complex systems as counting of very large quantities and keeping track of multiple events by cascading several counters. PLC counters also have the ability to control an unlimited number of contacts in a program just like timers.

Counter Instructions:

Counters are output instructions that provide the same functions, as would hardware counters. They are used to activate or de-activate a device after an expired count. A counter counts the occurrence of an event. The counter instructions require an accumulator register to store the elapsed count and a preset register to store a preset value. The preset value will determine the number of events occurrences that are to be counted. When the accumulated value equals the preset value, a status bit is net ON and can be used to turn on an output device.

PLC counters have programming formats similar to timer formats. One input furnishes count pulses which the PLC function analyses. Another input carries out enable-reset, as it does in the timer function.

Most PLC's include both down counters and up counters that function in similarity. The up-counter from zero upto preset count where some action takes place like counting the products produced. The down counter goes from a preset number down to zero, when the occurs like counting the raw materials used. Having both up and down counters enables a common resister to keep track of a net count.

CHAPTER - 5

APPLICATION OF MICROPROCESSOR BASED PLC

Application of microprocessor in any given relay logic can be transformed into a PLC program. Two sample systems were considered for demonstrating the working of the developed system. The system performance was tested and satisfactory operation was recorded.

5.1. Pump - Motor Circuit:

A simple pump-motor start - stop circuit is shown in fig 1-a. It is redrawn using the systems nomenclature of inputs (0- through 7) and outputs (8 through F) in the relay ladder format. The PLC program for this relay logic is given below:

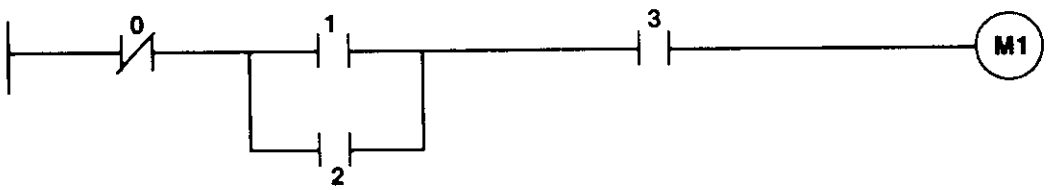
```
START:   LDC  0
         STO  TEMP
         LD   1
         OR   2
         AND TEMP
         AND  3
         OUT  8
         JMP  0
```

The test circuit for this pump - motor start - stop control is shown in fig 1.c.

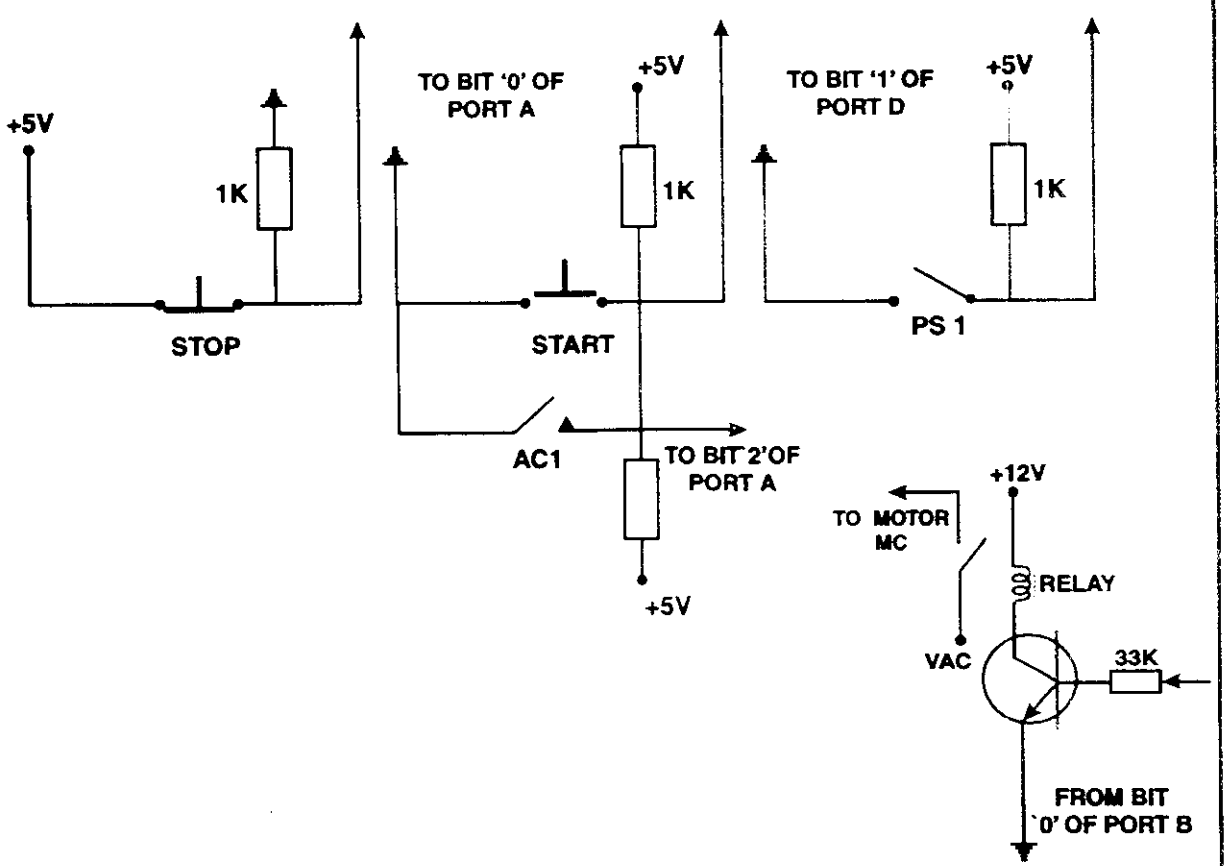
5.2. Automatic Mixer:

Fig 2-A is a sketch of an automatic mixing system. It shows a tank which is filled with a fluid, agitated and then emptied. Fig 2-B graphically illustrates the complete system cycle. In fig 2-C is given the ladder diagram for this application. Referring to fig 2-A to start push button (1PB) is pressed. This energizes a control relay (1CR) located in the start - stop box. Referring to fig 2-C when the float switch (1FS) is in the empty position and contact 1CR

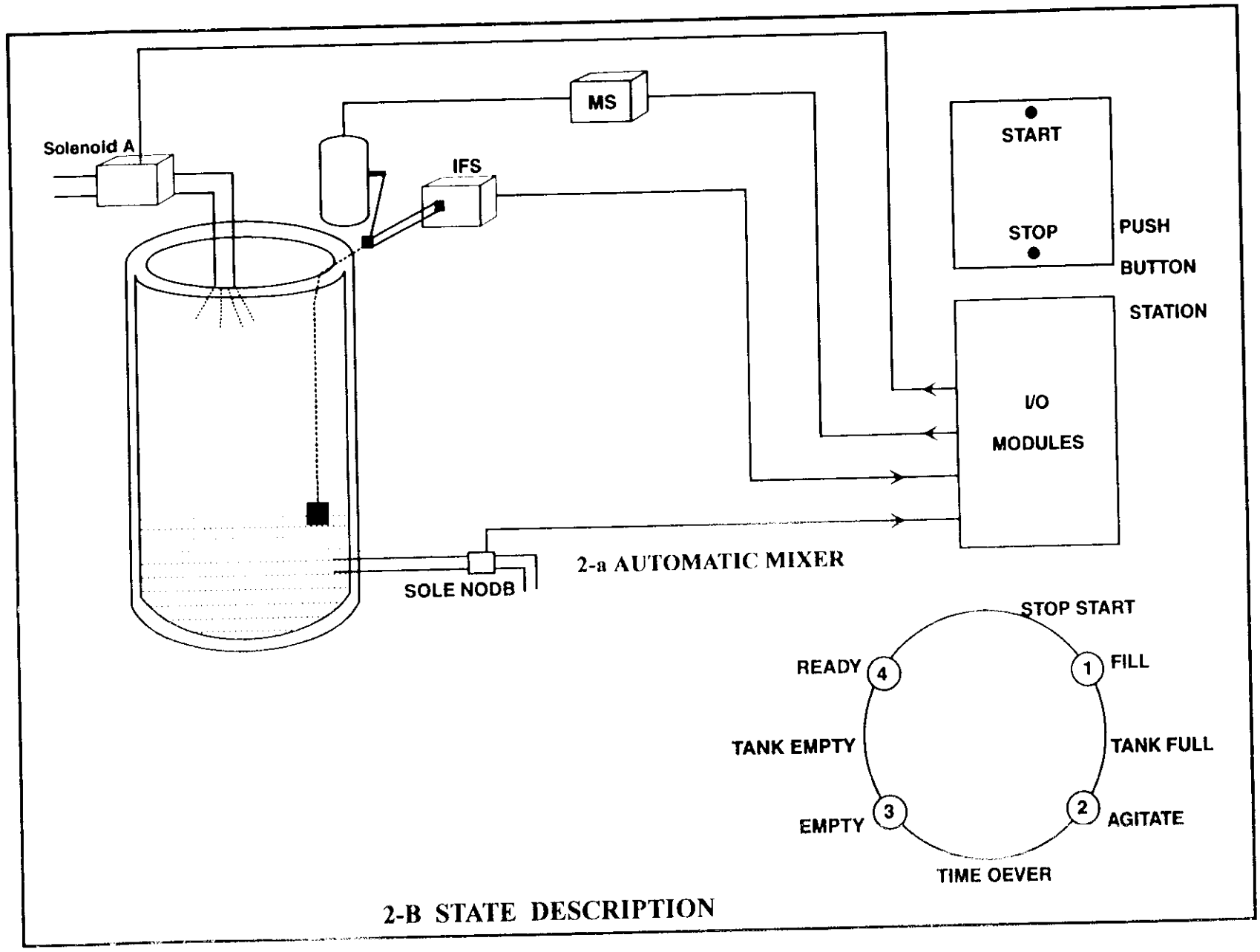
1.a. PUMP MOTOR CIRCUIT



1.b. PLC LADDER DIAGRAM FOR PUMP MOTOR CIRCUIT



1-C TEST CIRCUIT FOR PUMP-MOTOR START - STOP CONTROL

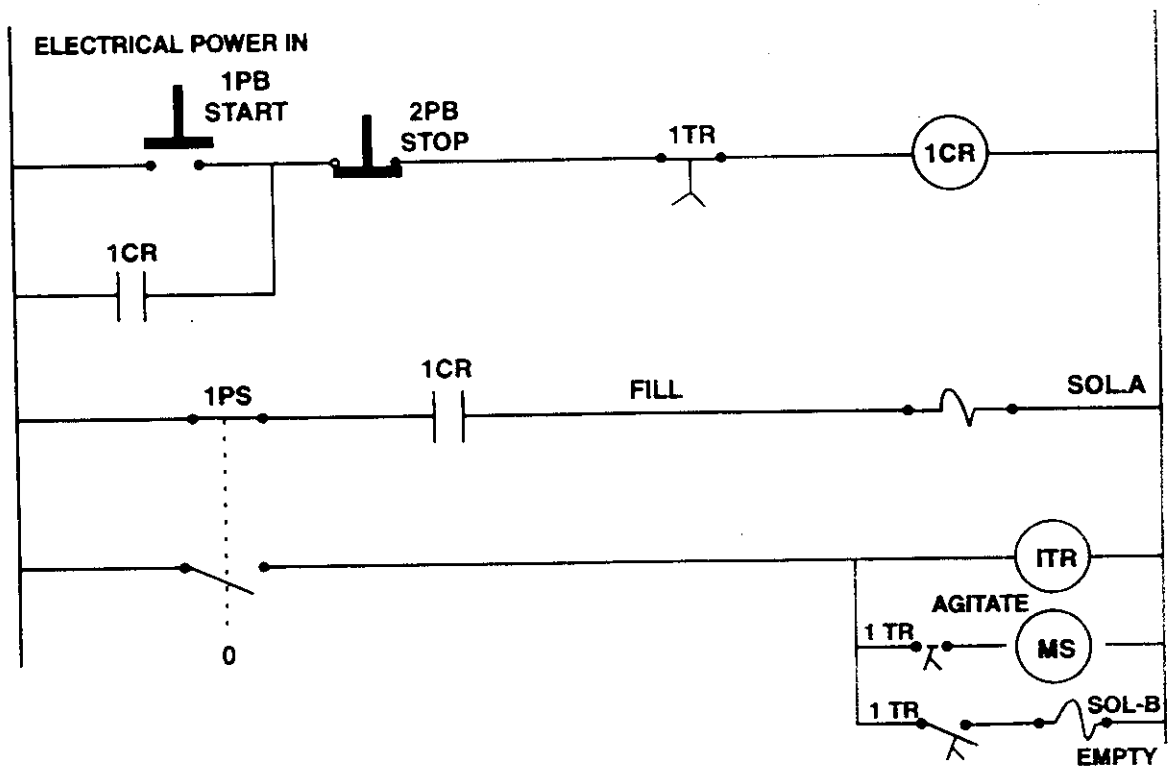


2-B STATE DESCRIPTION

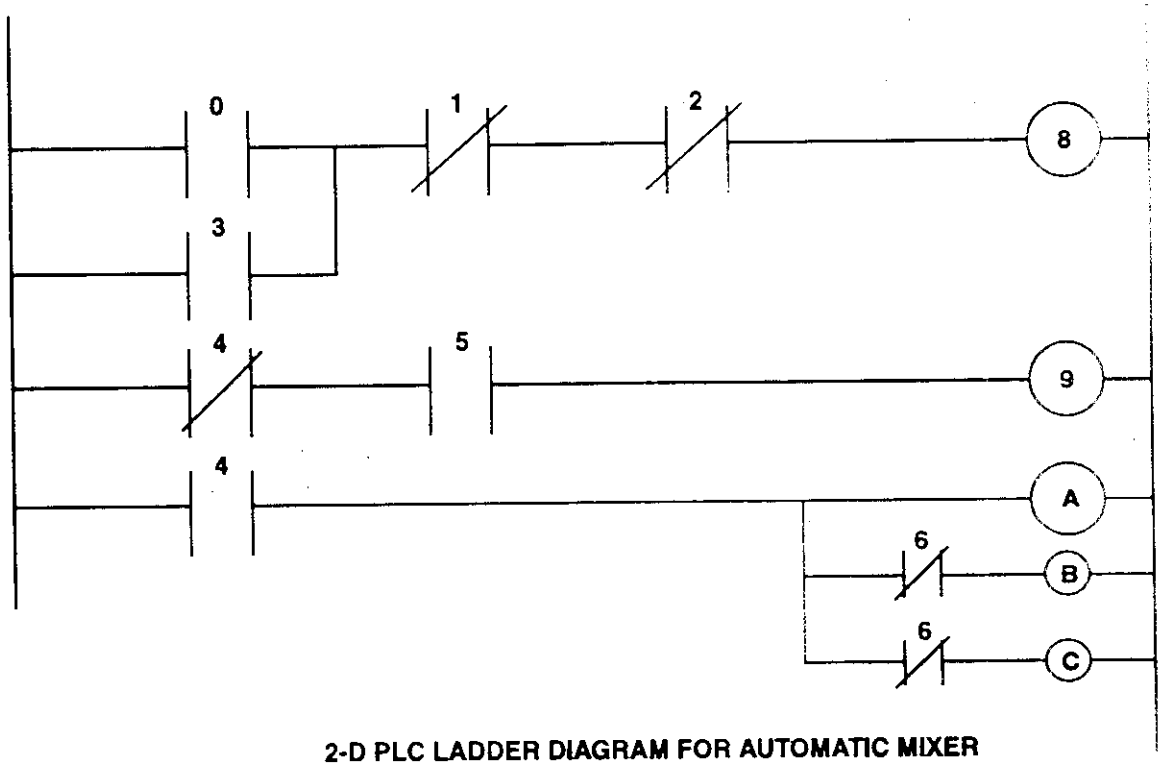
is closed. Solenoid A is energised, allowing fluid to fill the tank. When the tank is filled, float switch (1FS) changes to the 'filled' position, de-energising solenoid A, starting timer relay (1TR) in start - start box, and operating the mixer (MS). After the allotted time, relay 1TR switches off mixer and energises solenoid B, emptying the tank. When the tank is empty, float switch (1FS) shuts off solenoid B and places the system in 'Ready' for the next manual start. The PLC program for the relay logic shown in Fig.2.d is given below.

```
START   :   LD    0
            OR    3
            ANDC  1
            ANDC  2
            OUT   8
            LDC   4
            AND   5
            OUT   9
            LD    4
            OUT   A
            ORC   6
            OUT   B
            LD    4
            AND   6
            OUT   C
            JMP   0
```

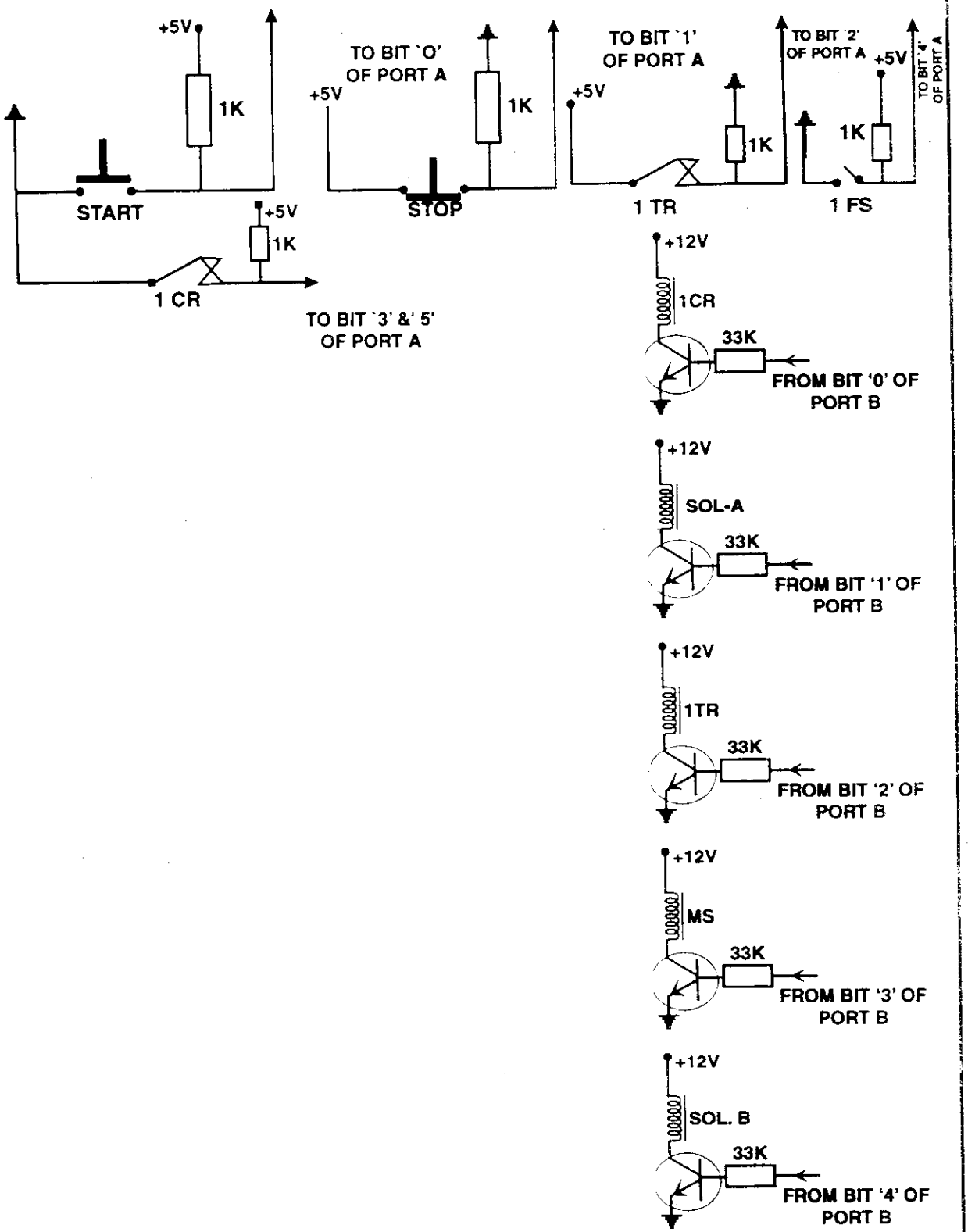
The test circuit for this automatic mixer control is shown in Fig.2.e.



2-C AUTOMATIC MIXER CONTROL



2-D PLC LADDER DIAGRAM FOR AUTOMATIC MIXER



TEST CIRCUIT FOR AUTOMATIC MIXER

Fig-2.e

CHAPTER - 6

APPLICATION OF PC BASED PLC

REPLACEMENT OF THE ELECTRONIC CONTROL OF GEAR HOBBING MACHINE BY PLC

6.1. Details of GHM:

6.1.1. Introduction:

Gears can be produced on small cylindrical brass piece of maximum dimension of 40mm by using this machine. The gears are produced due to the rotary movements. The front view of the machine where hobbing is done is shown in figure (1).

The work piece rotates along the spindle that is rotated by the main motor. While the work piece is rotating, the hob which is also rotating comes in contact with the work piece at the lest end and makes a horizontal travel towards right. Due to this simultaneous and relative motion of the hob and work piece, gears are produced on the work piece. Though both the spindle and hob are run by the same motor, speed of hob might vary according to the width of the gear to be produced. This speed variation is achieved by a clutch - gear arrangement.

6.1.2. Main Drives and Main Parts of GHM:

Main Drives Of GHM:

(1) Main motor

(2) Hydraulic motor

(3) Coolant pump

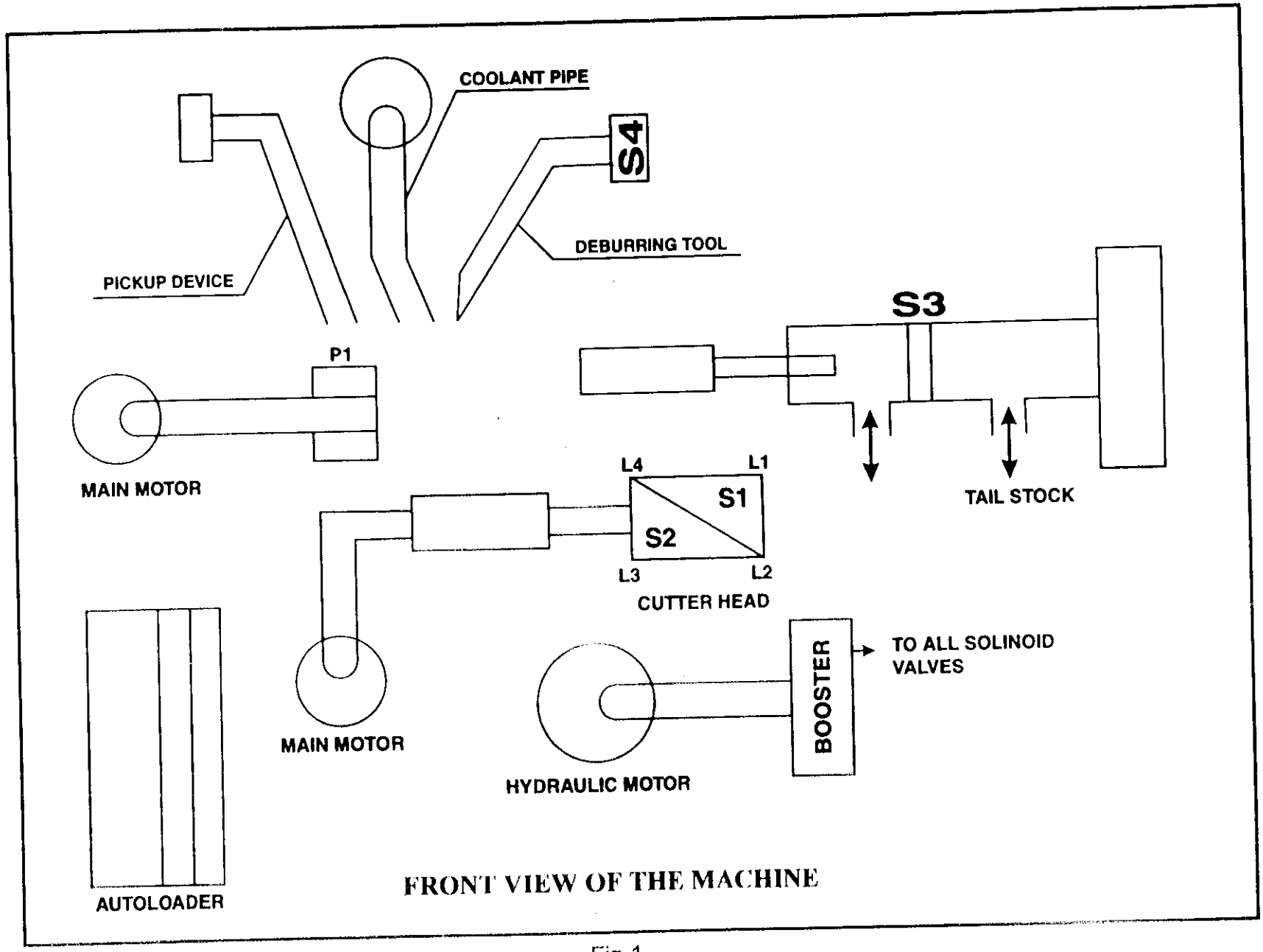

COOLANT PIPE
S4
DEBURRING TOOL
PICKUP DEVICE
S3
P1
MAIN MOTOR
TAIL STOCK
L4
L1
S1
S2
L3
L2
CUTTER HEAD
BOOSTER
TO ALL SOLINOID VALVES
MAIN MOTOR
HYDRAULIC MOTOR
AUTOLOADER
FRONT VIEW OF THE MACHINE

Fig-1

Main Motor:

It is squirrel cage Induction Motor of capacity of 0.75 Kw. This rotates the spindle and the hob. The main motor is used for the rotation of the hob and the work piece. The main motor output is applied to a gear box unit and is then given to the hob. The spindle is rotated at a constant speed irrespective of number of teeth and their dimensions, where as hob is rotated at an adjustable constant speed by using gear arrangement. The speed of rotation of hob is same as that of the main motor.

Hydraulic Motor:

It is also a Squirrel Cage Induction motor of capacity 0.08 Kw. This develops the hydraulic pressure, which is used to provide linear movements through out the machine. The output of hydraulic motor is applied to the pump, which is in turn connected to the solenoid valves. The hydraulic pump controls the vertical movement of the hob. A booster or accumulator is also provided to maintain a constant pressure. When the required pressure is developed by hydraulic motor on the fluid, a limit switch is operated. Further operations will be discontinued if this pressure limit is not satisfied.

Coolant Pump:

This is a motor pump arrangement. The capacity of the motor is 0.11 Kw. During hobbing operation heat is developed in the work piece which may damage the machine. In order to avoid over heating and damage, coolant is poured on the work piece and the coolant is reused again. A "float" is provided in order to have a control over the coolant level. If the coolant is below the required set level then the operation is discontinued.

Main Parts of A GHM:

(1) Spindle and tailstock

(2) Cutter head or hob

(3) Debarring tool

(4) Auto loader arrangement

Spindle and Tailstock:

They are used to hold the work piece. During the hobbing operation the work piece is rotated by rotating the spindle. The linear (to and for) movement of the tailstock is by using a solenoid.

Cutter Head:

The cutter head or hob does the hobbing operation. The speed of iinear movement of the hob is rapid in the beginning and later it is reduced. The speed variation at different times in the same cycle is achieved by using timers.

Debugging Tool:

This is used to remove the burr produced during hobbing. The debarring tool is moved by using the solenoids. This tool starts working after a few milliseconds after the hobbing has started.

Auto Loader Arrangement:

Loader is a conveying rail which has an attachment for loading the blank pieces individually. It is hydraulically operated. The auto loader system has got a motor of capacity of 0.032 Kw. It is a three phase Induction Motor rated for 2600 r.p.m. In the auto loader system, the work piece is made to move around the groove provided on the auto loader drum. The work piece is taken by the pick-up device

and placed in the work area. The pick - up device is provided with robot fingers to facilitate easy pick - up of the material.

6.1.3. Controllable Movements of GHM:

(1) Movement of pick-up device of auto loaders

(2) Movement of tail stock

(3) Vertical and horizontal movement of cutter head

(4) Debarring tool

Movement of Pick-Up Device:

The pick-up device moves from original position to the auto loader drum and picks up the raw material. From there it goes to work place and then it returns to the original position after keeping the raw material at the work place.

The three positions of the pick-up device are:

(a) At the home place (original place)

(b) At the auto loader drum

(c) At the work place

Movement of Tailstock:

It just makes to and for-motion to hold and release the work piece. The distance of the movement depends upon the length of the work piece.

Movement of Cutter Head:

The possible movements are:

(1) Up and Down (Vertical)

(2) Right to left and vice versa (horizontal)

Movement of Debarring Tool:

The debarring tool moves from the original position to the work place during the hobbing operation.

6.1.4. Working Sequence of the GHM:

The entire working sequence can be shown in single diagram as shown in the figure 2

(1) First the mains are switched on

(2) Next the hydraulic motor is switched on.

(3) Next the main motor is switched on.

(4) Then the raw material is loaded using the auto loader arrangement.

(5) The presence of raw material is checked using a proximity switch and if it is there the tailstock is moved forward.

(6) Immediately the coolant flow starts using a coolant pump.

(7) Cutter head moves upwards and as soon as the upper limit is reached cross travel starts. A timer (timer 1) is energised. When the right end is reached i.e. when hobbing is over a downward movement of the cutter head is preformed and then go to it's original position.

(8) The debarring tool is started when the timer 1 reads three seconds i.e. debarring lags behind cutter head by three seconds.

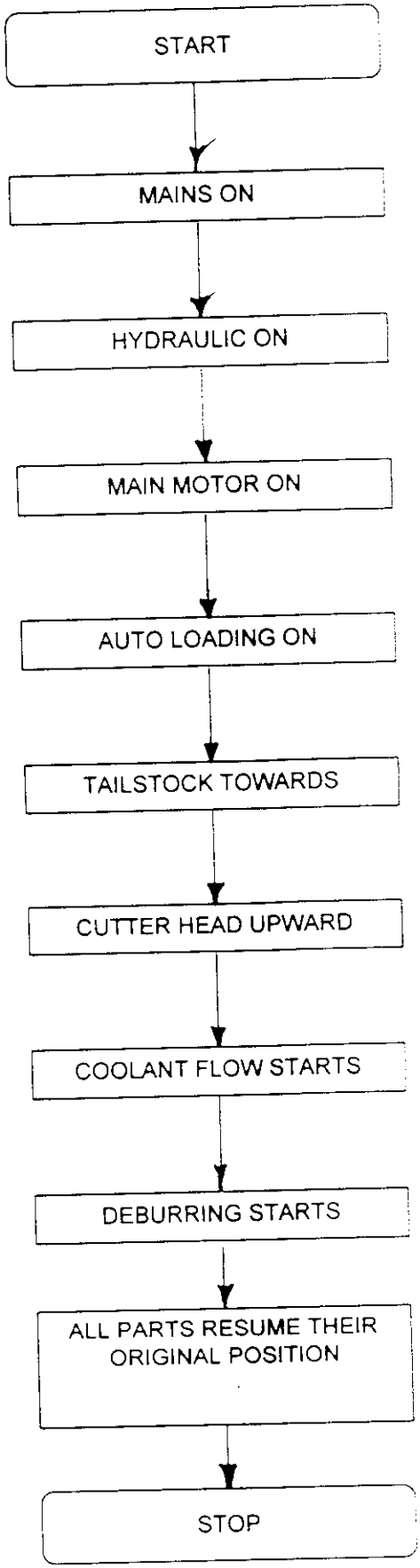
(9) After completing hobbing and debarring, the cutter head. the debarring tool and the tails stock resume their original position.

6.1.5. Controlling Devices for GHM:

(1) Selector switches

(2) Solenoids

(3) Limit switches



Work Sequence of GHM

Fig.2

(4) Proximity switches

a. Selector Switches:

A selector switch enables the operator to predetermine the manner in which his machine is to operate. The selector switch has two main parts, the mechanical actuator and the contact block.

Selector switches are of two types. They are:

(1) Push button

(2) Thumb wheel

Push Button:

It is a pilot device, which provides the control of an equipment by pressing an actuator, which looks like a button. Push button switches can be divided into two parts. One part is the mechanical actuator or button assembly and the second part is the electrical portion or contract assembly called the contact book.

To avoid confusion pushbutton are preferred to thumb wheel switches.

Note:- Colour of the button is also an important factor in push button switches.

Example: Red push buttons are used to stop and for emergency step operations, while green pushbuttons are used for start operations.

Thumb Wheel Switches:

This is used to replace so many push buttons as selector switches by a single multiselector switch for various devices. This device basically converts a decimal number to its equivalent binary number.

The different movements of various devices will be allotted a unique decimal number. Whenever we select a number by using thumb wheel rotation the corresponding device will perform the assigned movement.

Selector switches are mounted on the panel board. The selector switches are provided in order to select any particular option among the given few alternatives.

The selector switches of gear hobbing machines are used to:

(1) To determine the mode of operation

(2) Selector switch for manual cycle

Selector switch to determine the mode of operation:

A selector switch enables the operator to predetermine the manner in which this machine is to operate.

(1) Automatic

(2) Semi-automatic

(3) Manual

(1) Automatic:

All the movements including the loading of the work piece will be done automatically without manual intervention.

(2) Semi-Automatic:

In this mode some operations will be performed automatically. The operation like loading and removing will be done manually. In this mode every cycle should be re-started manually.

(3) Manual:

In this mode all the operations will be performed manually by selecting the appropriate selector switches. The machines will be operated in this mode in the rare cases like after doing some maintenance on that machine. The machine operation will be checked manually for a few cycles in order to avoid damages.

Selector Switches for manual cycles:

(1) Tail Stock Closing:

This selector switch is used in both manual cycle and semi-automatic. In the semi-auto mode, during each complete cycle the remover and feeding of work piece is done by manually operating the switch.

(2) Cutter Head Vertical Sliding:

This is only in manual mode. To select the cutter head movement vertically either in upward or down ward direction.

(3) Cutter head horizontal sliding:

(4) Engagement and disengagement of debarring tool:

Solenoid:

Solenoid is an important component in this machine. This is used for control purposes. It is an electromagnet that applies a straight line force in a push or pull motion. It contains a winding of insulated conductor and a plunge. The solenoid is shown in the figure 3.2

When the coil is energised the plunger is attracted towards it. So the pressurized liquid in area A1, comes out through port 1 and the pressurised liquid enters chamber A2 through port 2. When the coil is de-energised due to liquid

pressure the plunger moves backward. Then the pressurised liquid from A2 comes out and enters A1. Thus to and for the plunger makes motion and hence any other device attached to it.

Typical applications include Brakes, Conveyors, Safety devices etc.

Solenoid Valves:

Solenoids are operated along with solenoid valves. Solenoid valves is used to obtain mechanical movement in machinery by a utilizing fluid or air pressure. A solenoid valve is used in liquid lines and is electrically closed opened or closed to control the flow of the hydraulic fluid. Solenoid coils open these valves when energised and allows these to be closed when de-energised.

Solenoid Valves of GHM:

S1 - Cutter head vertical movement

The up and down movements of cutter head are made by energising and de-energising the coil of S1 through a control relay..

S2 - Cutter Head Horizontal Movements:

The left and right movements are made by energising and de-energising the coil of S2 respectively through another relay.

S3 - Tailstock:

The horizontal to and for motion of the burring tool is by energising and de-energising the coil of S3 through the control relay.

S4 - Debarring Tool:

The to and for motion of debarring tool is by energising and de-energising the coil of S4 through a relay.

Limit Switches:

Limit switches is an important control element. They are generally the mechanical sensors used to detect and control the linear or rotary movement of any moving part. They are also used to stop a mechanical movement of a machine and may also be used to stop a particular movement and initiate another movement.

The simple application of limit switch is in producing automatic to and fro movement of a planar machine bed.

Limit Switches In GHM:

Hydraulic Pressure Limit

Throughout the machine all the linear movements are performed only due to the hydraulic pressure developed by the motor. Only when the pressure of the fluid reaches a predetermined valve the functions should begin. To exert this control the pressure limit is used.

Coolant or Lubricating Oil Level:

To avoid mal operation and damage there should be continuous lubrication and cooling which can be ensured only when enough amount of oil is present in the reservoir. In order to ensure the correct liquid level limit switch is used.

Proximity Switches:

It is magnetic sensor. The proximity switch gets closed when a metallic part arrives near it.

Proximity Switches in GHM:

(1) Cutter head upper limit

A proximity is provided at the upper limit of cutter head movement

(2) Cutter head lower limit

(3) Cutter head right end limit

With above brief details about GHM the operations can be drawn in the ladder form.

6.1.6. Control Panel of GHM:

Components of the Control Panel:

The major components present in the control panel comprises of 2 transformers, contractors and relays. The lay out of the panel board is shown in the figure.

(1) TRANSFORMERS

(2) RELAYS

(3) CONTACTORS

Transformers:

As mentioned, the control panel consists of 2 transformers. The 1st transformer is used to step down 440V a.c. to 220V a.c. which is used to drive the motors. The 2nd transformer further steps down 220V a.c. to 24V a.c. which is then rectified using a rectifier and fed to the PLC.

Relays:

These relays consist of a bi-metallic strip which is heated by means of a heater coil. This coil is supplied through the current transformer. An insulated arm carrying the contact is pivoted and is held in contact with the strip through a spring

S, the tension of the spring is varied by sector shaped plate A. Under normal working conditions the strip remains straight but when the strip is heated to a predetermined temperature it bends. The tension of the spring is released and the contacts are closed. Thus the trip circuit is energised.

The thermal overload relay is shown in figure.4

These relays are connected in series with the contactor to provide effective protection. When the relay senses the over load it sends a signal to the contactor which opens its contacts. The thermal over load relay is to be used in association with a contactor in figure-5 for motor protection.

Contactors:

Contactors are mechanical switching devices. They consist of current carrying contacts. The contactors are capable of making or braking circuits as the need arises the contacts used here are normally engaged under predetermined conditions open to interrupt the circuit.

The contactor has one main contact and two auxiliary contacts. The main contact is used in switching ON and OFF the power circuits. Auxilliary contacts are used in control circuit.

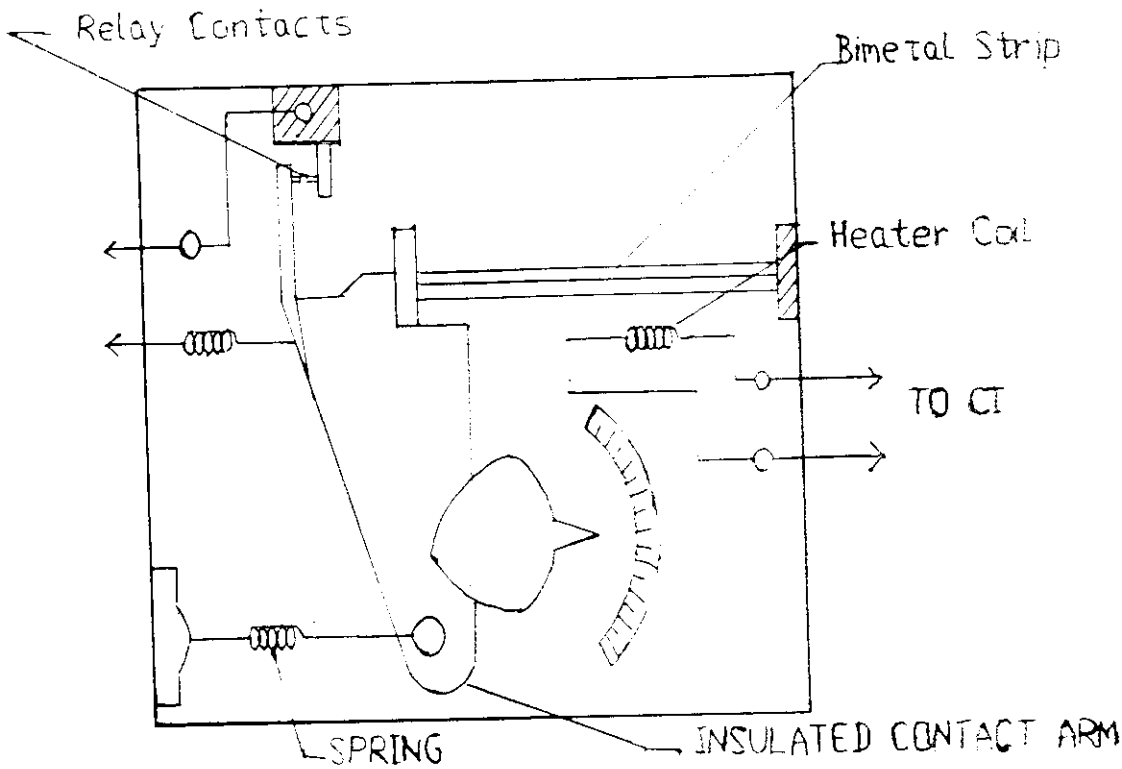


Figure 4 THERMAL CONTACT RELAY

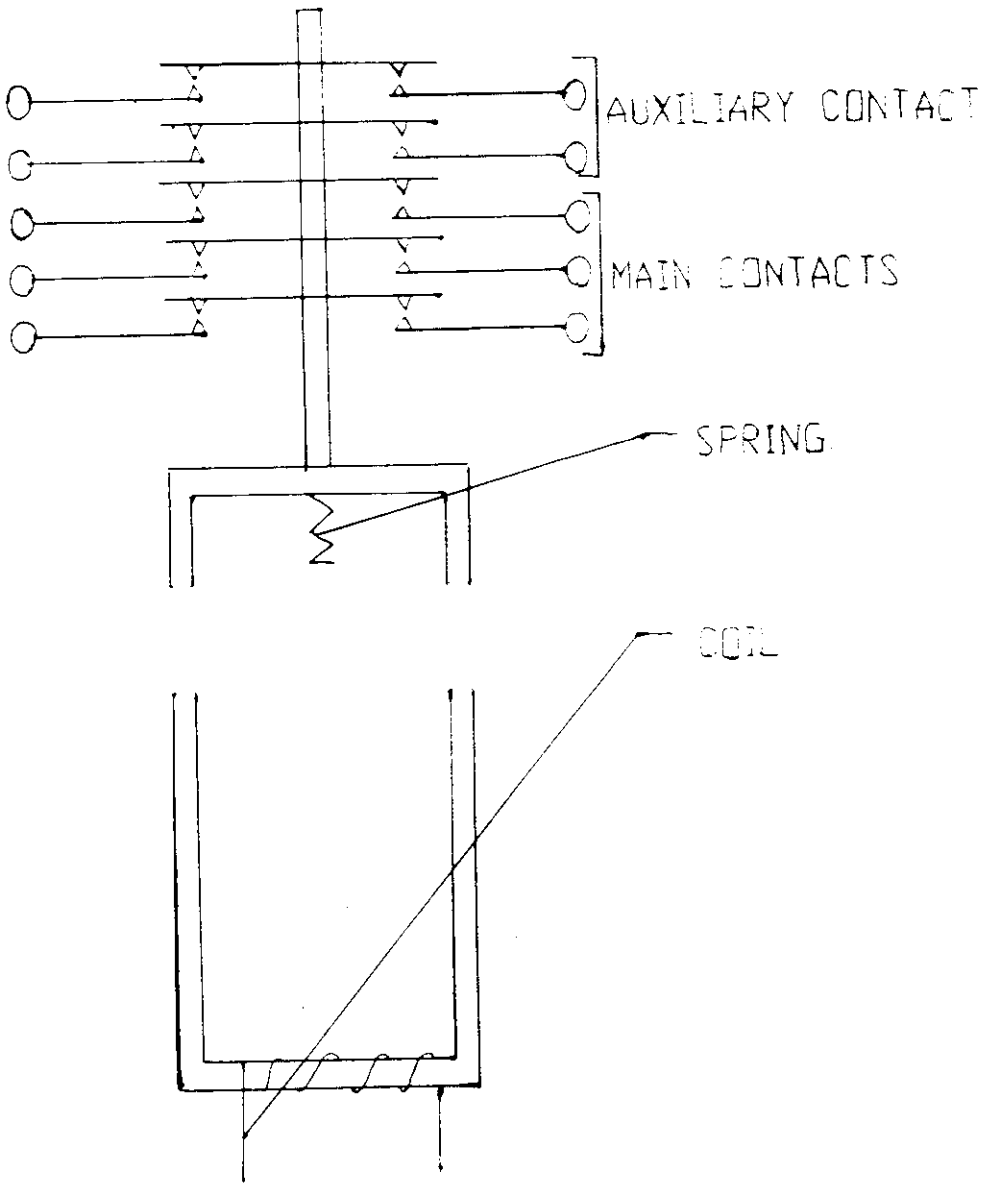


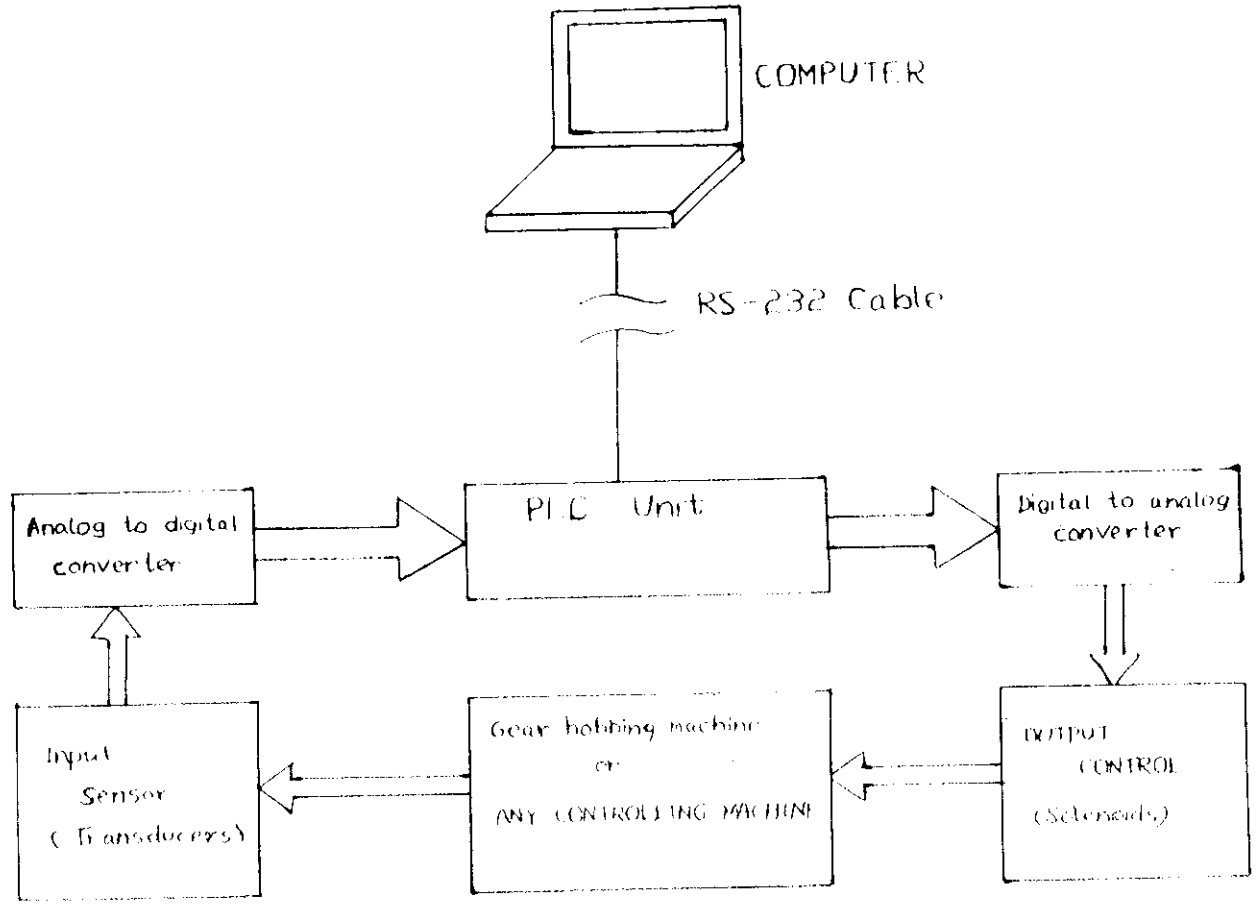
Figure 5 CONTACTOR

6.2. PLC in Gear Hobbing Machine:

Gear hobbing machine use the PLC. The software and hardware part of the PLC is as same as that of the microprocessed based PLC. There occurs a slight difference in the operation. The function of PC based PLC is described as follows.

The construction and operation of GHM is described in this section. In this section we are going to look at way of communicating the computer with PLC unit and how they can be connected to gear hobbing machine / any other controlling machine (i.e., transducer and solenoids). The basic control and communication sequence of PLC unit is shown in fig1.

The general control program is a program in which all the control operation in PLC is drawn into the ladder diagram and also the computer is connected with PLC through all control program and also the computer is connected with PLC through RS-232 cable. In PLC unit process the control program and give the necessary control signal to the PLC memory. The CPU in PLC unit process the control program and give the necessary control signal to the required output device. The control operation of GHM is sensed by the Input transducer, the output signal of the transducer is usually analog in nature. So this analog signal is converted in to digital signal by analog to digital converted. This digital signal is activated the control program which is already loaded into PLC memory. Thereafter, the PLC gives the necessary control signal to the required output device. Normally, PLC output signal is a digital signal. This digital signal is converted into analog signal by signal is activated the solenoid which is used to give the mechanical action to the gear hobbing machine or any other controlling machine. The gear hobbing machine ladder diagram is given as follows:

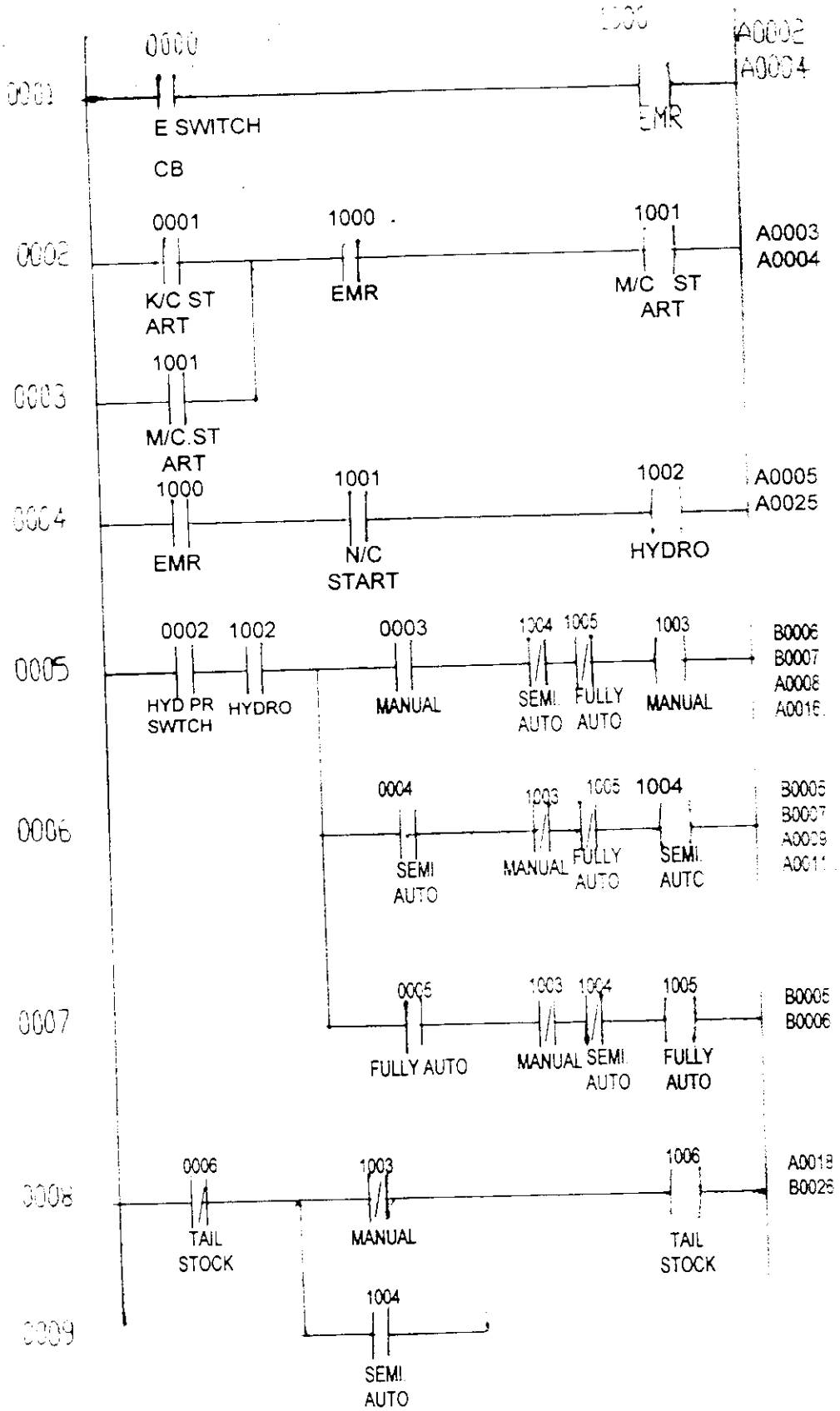


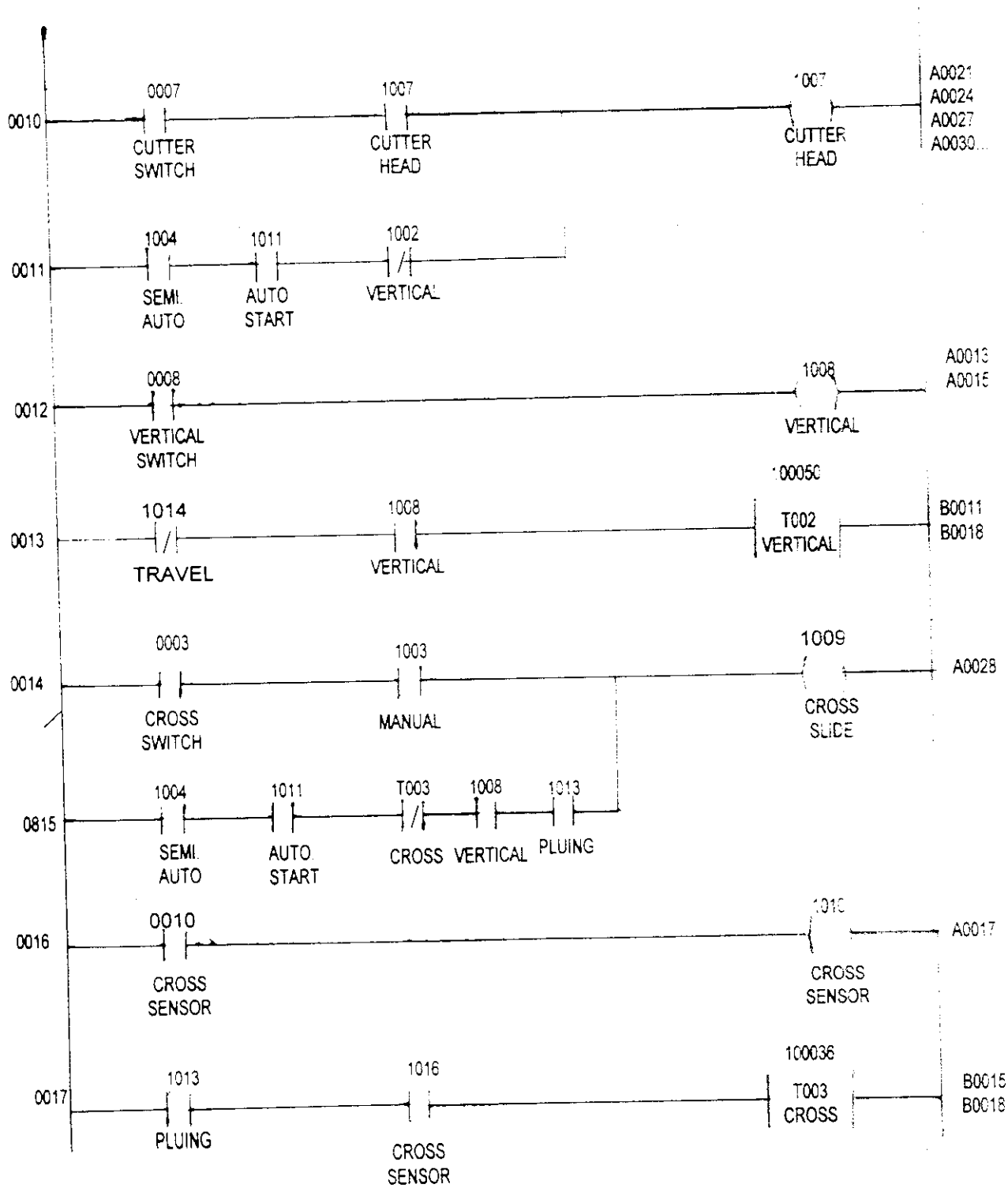
Basic Control and Communication Sequence of PLC

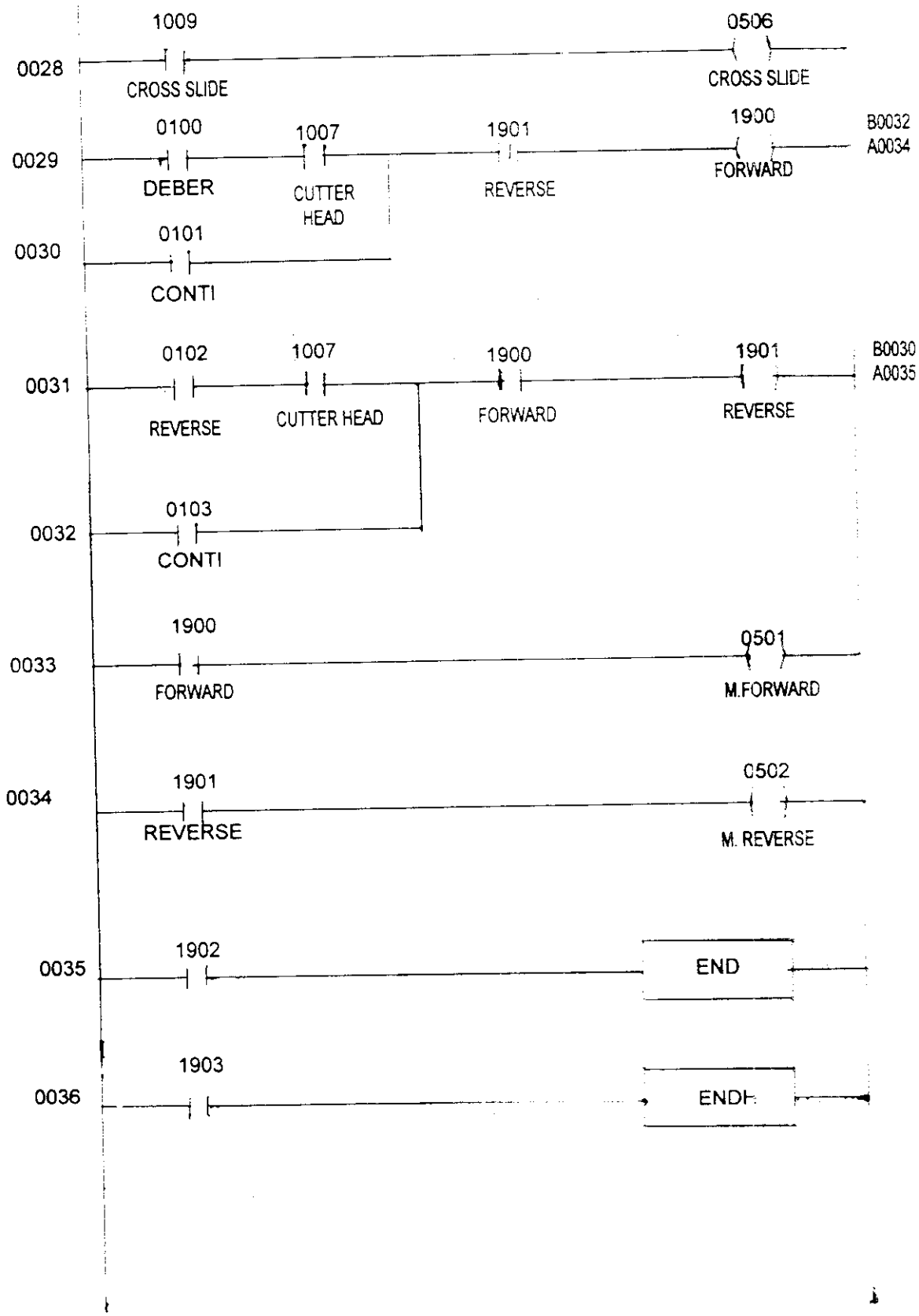
Fig - 1

PROGRAM BASED ON WHICH THE GEAR HOBBING MACHINE WORKS

Machine Type : KV40/80 (LADDER GRAPH) 99/08/10 Page (1
File Name : W81-Raj Title : WBLI81




0010
0007
CUTTER SWITCH
1007
CUTTER HEAD
'007
CUTTER HEAD
A0021
A0024
A0027
A0030...
0011
1004
SEMI AUTO
1011
AUTO START
1002
VERTICAL
0012
0008
VERTICAL SWITCH
1008
VERTICAL
A0013
A0015
0013
1014
TRAVEL
1008
VERTICAL
'00050
T002
VERTICAL
B0011
B0018
0014
0003
CROSS SWITCH
1003
MANUAL
1009
CROSS SLIDE
A0028
0815
1004
SEMI AUTO
1011
AUTO START
T003
CROSS
1008
VERTICAL
1013
PLUING
0016
0010
CROSS SENSOR
1010
CROSS SENSOR
A0017
0017
1013
PLUING
1016
CROSS SENSOR
100036
T003
CROSS
B0015
B0018


0028
1009
CROSS SLIDE
0506
CROSS SLIDE
0029
0100
DEBER
1007
CUTTER HEAD
1901
REVERSE
1900
FORWARD
B0032
A0034
0030
0101
CONTI
0031
0102
REVERSE
1007
CUTTER HEAD
1900
FORWARD
1901
REVERSE
B0030
A0035
0032
0103
CONTI
0033
1900
FORWARD
0501
M.FORWARD
0034
1901
REVERSE
0502
M. REVERSE
0035
1902
END
0036
1903
END

CHAPTER - 7

PLC ADVANTAGES AND DISADVANTAGES

7.1. PLC Advantages:

a. Flexibility:

In the past, each electrically controlled production machine required its own controller; 15 machines might require 15 different controllers. Now it is possible to use just one model of a PLC to run anyone of the 15 machines. Furthermore. less than 15 controllers can also be used, because one PLC can easily run on any machine. Each of the 15 machines under PLC control would have it's own district program.

b. Implementing changes and correcting errors:

With a wired relay-type panel any program alterations require time for receiving of panels and devices. When a PLC program circuit or sequence design change is made, the PLC program can be changed from a keyboard sequence in a matter of minutes. No revising is required for PLC controlled systems. Also, if a programming error has to be corrected in a PLC control ladder diagram; a change can be typed in quickly.

c. Lower Cost:

Increased technology makes possible to condense more functions in to smaller and less expensive packages. Now, we can purchase a PLC with a numerous relays, timers and counters, a sequencer and other functions for a few hundred dollar A PLC programmed circuit can be prerun and evaluated in the office or lab. The program can be typed in tested, observed and modified if needed, saving valuable factory time.

i. Documentation:

An immediate print out of the true PLC circuit is available in minutes if required. There is no need to look for the blue print of the circuit in remote files. The PLC prints out the actual circuit in operation at a given moment. A PLC print out is the circuit at the present-time, no wire tracing is needed for verification.

j. Sensitivity:

A PLC program change cannot be made unless the PLC is properly uniocked and programmed.

k. Ease of changes by re-programming:

Since the PLC can be reprogrammed quickly; mixed production processing can be accomplished. For example, if part B comes down, the assembly line while part A is still being processed, a program from part B's processing can be re-programmed into the production machinery in a matter of seconds.

7.2. PLC Disadvantages:

a. New Technology:

It is difficult to change the thinking of some personnel from ladder and relays to the PLC computer concept.

b. Fixed program applications:

Some applications are single-function applications. It does not pay to use a PLC that includes multiple programming capabilities if they are not needed.

c. Environmental considerations:

Certain process environments such as high heat and vibration, interfere with the electronic in PLC's which limit their use.

d. Fail-Safe Operation:

In many systems, the stop button electrically disconnects the circuit; if the power fails, the system stops. Further more, the relay system automatically restart when the power is re-stored. This of-course can be programmed into the PLC; however in some PLC programs we have to apply an input voltage to cause a device to stop. There the systems are not fail-safe. This disadvantage could be over-come by adding safety relays to the PLC system.

e. Fixed Circuit Operation:

If the circuit in operation is never altered, a fixed control system (such as a mechanical drum) might be less costly than a PLC. The PLC is most effective when periodic changes in operation are made.

CHAPTER - 8

CONCLUSION

PLC has become heart of todays manufacturing and process operations:

A microprocessor - based programmable controller to handle a maximum of 8 inputs and 8 outputs has been designed and developed. The system hardware including keyboard and I/O simulator have been fabricated. The software for keyboard scanning and conversion program have been developed and debugged. A few control application programs were tested.

Further Scopes:

Typically inputs to the system are the control-signals from the process stages and/or machines. For the prototype testing these signals were simulated by means of switches. Better would be if input interface units can be designed so as to accept typical industrial signal. Further, to avoid noise effects optical couplers can be used for isolation. Similarly output interface units can be used to raise the logic levels of the PLC output signals to the required process or machine voltage levels. The software can be extended to perform counting, arithmetic A/C and D/A conversion and thereby cater to almost all types of practical system requirements.

Regarding to replacement the electronic control in a GHM by PLC, presently the machine is working on manual and semi-automatic mode. This has greatly enhanced the gear production comparatively. It saves time and does not require much of manual labour and attention.

CHAPTER - 9

APPENDIX - A

EC-85 System Description:

The EC-85 trainer kit used in this project is an 8085 based microcomputer system. It is designed specifically for importing training in microprocessor applications. The unit facilities development of microprocessor based system by extending the memory capacity and by interfacing suitable for more than 10 devices.

The system can operate in two modes, viz (i) keyboard mode and (ii) serial input / output device mode (SIOD).

In keyboard mode the commands are entered via -The HEX keyboard and two system responds by displaying the desired information on the 7-segment LEN display.

In SIOD mode the commands are entered via one of the following 40 devices: Teletype, CFT terminal and tele printer.

Microprocessors kit operates at a clock rate of 3MHz devised internally by dividing a 6MHz crystal frequency by a factor of 2. The system has an 8 bit address latch(8212) which is used to demultiplex the lower order address information. The system has a 3 to 8 line decoder (74LS138.1) to proved chip select signals to varies 40 and peripheral ICs in new system and another decoder (74LSI38.2) for the memory.

The system monitor program is stored in EPROM (271287/2764). An optional socket is available and can be used to put a user EPROM (2764/2732) (62128/62256) RAM is provided for user applications.

A counter / timer (8253) is also provided on board of the three 16-bit counters contained in his chip, counter 'O' is used in the systems for single instruction capability and counters '1' and 2 are available to the user.

The system has a keyboard / display controller (8279) which interface to the system on one side and the keyboard and the display on the other side. This chip does keyboard scanning and display refresh without CPU intervention.

The programmable peripheral interface chips (8255) with TTL compatible, bi-directional I/0 lines are available for the user.

APPENDIX-B
SOFTWARE LISTING
FOR
"MICROPROCESSOR BASED PROGRAMMABLE CONTROLLER"

```
                        ;MONITOR STARTS FROM LOC,0800H
                        SYSTEM EMPLOYS SOFTWARE METHOD OF KEY SCANNING

                        ;INITIALISATION ROUTINE

0800                    ORG 0800H           ;RESERVE AND CLEAR FOUR
279D=                   ADRES EQU 279DH     ;RAM LOCATIONS FOR USE
279F=                   DATA  EQU 279FH     ;AS ADDRESS REG AND
2795=                   MONSP EQU 2795H     ;DATA REG
06E4=                   DGDISP EQU 06E4H
06CD=                   ADDISP EQU 06CDH
0800 219D27 MONST:      LXI H,ADRES
0803 0604               MVI B,04H
0805 AF                 XRA A
0806 77     PHI:        MOV M,A
0807 23                 INX H
0808 05                 DCR B
0809 C20608             JNZ PHI

                        ;KEY SCAN ROUTINE STARTS

080C 3E90   START:      MVI A,90H       ;PROGRAM 8255-1 AS
080E D303               OUT 03H         ;A-I/P,B-I/P,C-O/p
0813 AF                 XRA A           ;CLEAR ACC.& CARRY
0810 319527             LXI SP,MONSP    ;INITIALISE MONITOR SP
0814 D301               OUT 01H         ;O/P 0 TO ALL COLS
0816 DB00   KYOPN:      IN 00H          ;READ ROWS AND CHECK
0818 FEFF               CPI 0FFH        ;IF ALL KEYS ARE OPEN
081A CA1D08             JNZ KYOPN       ;TO ELIMINATE TWO KEY
                                        ;ROLLOVER
081D DB00   KYCLS:      IN 00H          ;READ ROWS AND CHECK
081F FEFF               CPI 0FFH        ;IF ALL KEY IS PREESED
0821 CA1D08             JZ KYCLS
0824 CD1909             CALL CHATER     ;SOFTWARE DEBOUNCE
0827 3EFE               MVI A,0FEH
```

```
0829 0F      NXTCOL:  RRC
082A 47               MOV B,A          ;SAVE COL.CODE
082B D301             OUT 01H          ;O/P 0(BIT 7) TO I COL
082D DB00             IN 00H           ;READ ROWS
082F 4F               MOV C,A          ;SAVE ROW CODE
0830 FEFF             CPI 0FFH
0832 C23908           JNZ DECODE       ;CHECK IF KEY PRESSEDIS
0835 78               MOV A,B          ;IN THIS COL OTHERWISE O/P
0836 C32908           JMP NXTCOL       ;ZERO TO NEXT COL
0839 78      DECODE:  MOV A,B          ;FROM SAVED COL CODE
083A FEFF             CPI 7FH          ;DETERMINE THE COL OF
083C CA4E08           JZ COL1          ;PRESSED KEY AND JUMP TO
083F FEBF             CPI 0BFH         ;RESPECTIVE COL ROUTINE
0841 CA6008           JZ COL2
0844 FEDF             CPI 0DFH


0846 CA7208           JZ COL3
0849 FEEF             CPI 0EFH
084B CA8408           JZ COL4

                      ;COLUMN ROUTINES
                      ;REALISATION OF KEY CODES

084E 213109  COL1:    LXI H,TAB        ;DEPENDING ON COL.CODE FIND
0851 160F             MVI D,0FH        ;ROW CODE MATCH IN TABLE
0853 79      NEXT1:   MOV A,C          ;IF MATCH FOUND REG.D HAS
0854 BE               CMP M            ;KEY CODE
0855 CA9608           JZ DONE
0858 23               INX H
0859 15               DCR D
085A 7A               MOV A,D
085B FE08             CPI 08H
085D D25308           JNC NEXT1
0860 213109  COL2:    LXI H,TAB        ;DEPENDING ON COL.CODE FIND
0863 1607             MVI D,07H        ;ROW CODE MATCH IN TABLE
0865 79      NEXT2:   MOV A,C          ;IF MATCH FOUND REG.D HAS
0866 BE               CMP M            ;KEY CODE
0867 CA9608           JZ DONE
086A 23               INX H
086B 15               DCR D
086C 7A               MOV A,D
086D FE00             CPI 00H
```

```
086F D26508           JNC NEXT2
0872 213109 COL3:     LXI H,TAB       ;DEPENDING ON COL.CODE FIND
0875 161F             MVI D,1FH       ;ROW CODE MATCH IN TABLE
0877 79     NEXT3:    MOV A,C         ;IF MATCH FOUND REG.D HAS
0878 BE               CMP M           ;KEY CODE
0879 CA9608           JZ DONE
087C 23               INX H
087D 15               DCR D
087E 7A               MOV A,D
087F FE18             CPI 18H
0881 D27708           JNC NEXT3
0884 213109 COL4:     LXI H,TAB       ;DEPENDING ON COL.CODE FIND
0887 1617             MVI D,17H       ;ROW CODE MATCH IN TABLE
0889 79     NEXT4:    MOV A,C         ;IF MATCH FOUND REG.D HAS
088A BE               CMP M           ;KEY CODE
088B CA9608           JZ DONE
088E 23               INX H
088F 15               DCR D
0890 7A               MOV A,D
0891 FE10             CPI 10H
0893 D28908           JNC NEXT4
0896 7A     DONE:     MOV A,D
0897 E610             ANI 10H
0899 CAB908           JZ DIGIT        ;CHECK & JUMP TO DIGIT
089C 7A               MOV A,D         ;ROUTINE IF KEY PRESSED
089D FE1C             CPI 1CH         ;IS DIGIT KEY
089F DACC08           JC INSTR        ;CHECK & JUMP TO INSTR
08A2 D61C             SUI 1CH         ;ROUTINE IF KEY PRESSED
08A4 1600             MVI D,00H       ;IS INSTR KEY
08A6 87               ADD A
08A7 5F               MOV E,A         ;IF NEITHER JUMP TO
08A8 21BI08           LXI H,COMTAB    ;RESPECTIVE COMMAND
08AB 19               DAD D           ;ROUTINE
08AC 7E               MOV A,M
08AD 23               INX H
08AE 66               MOV H,M
08AF 6F               MOV L,A
08B0 E9               PCHL
08B1 1409   COMTAB:   DW TEMP
08B3 1009             DW RUN
08B5 E508             DW ADRSET
08B7 FD08             DW MEMWR
08B9 CD2609 DIGIT:    CALL SHIFT      ;TO MONITOR SUBROUTINE SHIFT
08BC 3A9F27           LDA DATA
```

```
08BF B2                 ORA D
08C0 329F27             STA DATA
08C3 32F627             STA 27F6H
08C6 CDE406             CALL DGDISP     ;DATA DISPLAY EC-85 MONITOR
08C9 C30308             JMP START       ;SUBROUTINE
08CC CD2609 INSTR:      CALL SHIFT
08CF CD2609             CALL SHIFT
08D2 3A9F27             LDA DATA
08D5 B2                 ORA D
08D6 329F27             STA DATA
08D9 32F627             STA 27F6H
08DC CDE406             CALL DGDISP     ;DATA DISPLAY EC-85 MONITOR
08DF CD2609             CALL SHIFT      ;SUBROUTINE
08E2 C30C08             JMP START
08E5 2A9F27 ADRSET:     LHLD DATA       ;(HL)=DATA REG
08E8 229D27             SHLD ADRES      ;SAVE (HL) IN ADRES REG.
08EB 3A9D27 DISPAD:     LDA ADRES
08EE 32F427             STA 27F4H
08F1 3A9E27             LDA 279EH
08F4 329527             STA 27F5H
08F7 CDCD06             CALL ADDISP     ;ADDRESS DISPLAY EC-85 MONITOR
08FA C30C08             JMP START       ;SUBROUTINE


                        ;MEMORY WRITE AND ADDRESS INCREMENT

08FD 2A9D27 MEMWR:      LHLD ADRES      ;(HL)=ADDR.REG
0900 3AA027             LDA 27A0H       ;ACC.=DATA REG.+1
0903 77                 MOV M,A         ;DATA WRITE I BYTE
0904 23                 INX H           ;ADDR.INC
0905 3A9F27             LDA DATA
0908 77                 MOV M,A
0909 23                 INX H
090A 229D27             SHLD ADRES
090D C3EB08             JMP DISPAD      ;JUMP TO ADDR.DISPLAY

                        ;MONITOR TO USER CONTROL ROUTINE

0910 2A9D27 RUN:        LHLD ADRES      ;(HL)=ADDR.REG
0913 E9                 PCHL            ;(HL)=(PC)
0914 161C   TEMP:       MVI D,1CH       ;COMMAND KEY TREATED AS
0916 C3B908             JMP DIGIT       ;DIGIT KEY WITH CODE 1C

                        ;MONITOR SUBROUTINES

0919 0608   CHATER:     MVI B,08H       ;DELAY FOR KEY DEBOUNCE
```

```
091B 0EFF     DOLLAR:  MVI C,0FFH
091D 0D       POUND:   DCR C
091E C21D09            JNZ POUND
0921 05                DCR B
0922 C21B09            JNZ DOLLAR
0925 C9                RET
0926 2A9F27   SHIFT:   LHLD DATA      ;SUBROUTINE FOR DATA SHIFT
0929 29                DAD H          ;BY FOUR BITS092A 29
092A 29                DAD H
092B 29                DAD H
092C 29                DAD H
092D 229F27            SHLD DATA
0930 C9                RET
0931 7F       TAB:     DB 7FH         ;ROW CODE MATCH TABLE
0932 BF                DB 0BFH
0933 DF                DB 0DFH
0934 EF                DB 0E7H
0935 F7                DB 0FBH
0936 FB                DB 0FDH
0937 FD                DB 0FEH
0938 FE                DB 0FEH

                       ;xxxxxxxxxxxxxxxxxxx
                       ;CONVERSION PROGRAM
                       ;xxxxxxxxxxxxxxxxxxx

0940                   ORG 0940H
2797 =                 CODEPG EQU 2797H
2795 =                 CONTPG EQU 2795H
278B =                 CONVSP EQU 278BH
05B5 =                 OUTPUT EQU 05B5H
03A0 =                 DLAY EQU 03A0H
032B =                 CLEAR EQU 032BH
0940 21020             LXI H,2000H    ;KEY CODES OF USER
0943 229727            SHLD CODEPG    ;STORED FROM 2000H
0946 210D22            LXI H,220DH    ;CONVERTED PROGRAM
0949 229527            SHLD CONTPG    ;STORED FROM 220DH
094C 318B27            LXI SP,CONVSP  ;SP INITIALISATION
094F 210022            LXI H,2200H    ;CONVERTED PROGRAM
0952 363E              MVI M,3EH      ;INITIALISATION BEGINS
0954 23                INX H
0955 3690              MVI M,90H      ;PROGRAM 8255-2 PORTS AS
0957 23                INX H          ;A-I/P,B-O/P,C-O/P
0958 36D3              MVI M,0D3H
095A 23                INX H
095B 360B              MVI M,OBH
095D 23                INX H
```

```
095E 3621                MVI M,21H      ;SAVE KEY CODE STARTING
0960 23                  INX H          ;ADDRESS IN CODEPG
0961 3600                MVI M,00H
0963 23                  INX H
0964 3620                MVI M,20H
0966 23                  INX H
0967 3622                MVI M,22H
0969 23                  INX H
096A 3697                MVI M,97H
096C 23                  INX H
096D 3627                MVI M,27H
096F 23                  INX H
0970 3631                MVI M,31H      ;CONVERTER PROGRAM STACK
0972 23                  INX H          ;POINTER INITIALISATION
0973 3686                MVI M,86H
0975 23                  INX H
0976 3627                MVI M,27H
0978 217C27              LXI H,277CH    ;ERROR ROUTINE
097B 360E                MVI M,0EH      ;INITIALISATION
097D 23                  INX H
097E 3614                MVI M,14H
0980 23                  INX H
0981 3615                MVI M,15H
0983 23                  INX H
0984 3614                MVI M,14H
0986 2A9727 SARTC:       LHLD CODEPG    ;CHECK CODE FOR INSTRUCTION
0989 7E                  MOV A,M        ;AND JUMP TO CORRESPONDING
0988 FE10                CPI 10H        ;INSTRUCTION ROUTINE
098C DA010C              JC ERROR
098F E60F                ANI 0FH
0991 0600                MVI B,00H
0993 87                  ADD A
0994 4F                  MOV C,A
0995 219E09              LXI H,INSTAB
0998 09                  DAD B
0999 7E                  MOV A,M
099A 23                  INX H
099B 66                  MOV H,M
099C 6F                  MOV L,A
099D E9                  PCHL
099E B609    INSTAB:     DW JUMP        ;INSTRUCTION TABLE
09A0 C509                DW LD
09A2 D509                DW LDC
09A4 EE09                DW AND
09A6 010A                DW ANDC
09A8 1708                DW OR
09AA 2A0A                DW ORC
```

```
09AC 460A              DW DELAY
09AE 6F0A              DW HALT
09B0 780A              DW OUT
09B2 8B0A              DW OUTC
09B4 B30A              DW STO


        ;REALISATION OF 8085 CODE ROUTINES FOR KEY CODES BEGIN

09B6 CD7D0B JUMP:      CALL KYPAS1    ;PROGRAM GENERATE 8085 CODES
09B9 2A9527            LHLD CONTPG    ;CORRESPONDING TO THE
09BC 36C3              MVI M, 0C3H    ;INSTRUCTION JMP 2204H
09BE 23                INX H
09BF 3604              MVI M,04H
09C1 23                INX H
09C2 3622              MVI M,22H
09C4 76                HLT
09C5 CD880B LD:        CALL KYPAS2    ;PROGRAM GENERATE 8085 CODES
09C8 CDC20B            CALL CONSIN    ;TO LOAD THE STATUS OF THE
09CB 23                INX H          ;SELECTED I/P & STORE
09CC 365F              MVI M,5FH      ;IN REG. E
09CE 23                INX H
09CF 229527            SHLD CONTPG
09D2 C38609            JMP STARTC
09D5 CD880B LDC:       CALL KYPAS2    ;PROGRAM GENERATE 8085 CODES
09D8 CDC20B            CALL CONSIN    ;TO LOAD THE STATUS OF THE
09DB 23                INX H          ;SELECTED I/P ,COMPLEMENT
09DC 362F              MVI M,2FH      ;IT & STORE IT REG.E
09DE 23                INX H
09DF 36E6              MVI M,0E6H
09E1 23                INX H
09E2 3601              MVI M,01H
09E4 23                INX H
09E5 365F              MVI M,5FH
09E7 23                INX H
09E8 229527            SHLD CONTPG
09EB C38609            JMP STARTC
09EE CD880B  AND:      CALL KYPAS2    ;PROGRAM GENERATE 8085 CODES
09F1 CDC20B            CALL CONSIN    ;TO LOAD THE STATUS OF THE
09F4 23                INX H          ;SELECTED I/P "AND" IT WITH
09F5 3683              MVI M,0A3H     ;THE CONTENT OF THE REG.E &
09F7 23                INX H          ;STORE THE RESULT IN REG.E
09F8 365F              MVI M,5FH
09FA 23                INX H
09FB 229527            SHLD CONTPG
09FE C38609            JMP STARTC
0A01 CD880B  ANDC:     CALL KYPAS2    ;"AND" THE SELECTED I/P
```

```
0A5F 23                 INX H
0A60 36CD               MVI M,0CDH
0A62 23                 INX H
0A63 3645               MVI M,45H
0A65 23                 INX H
0A66 3603               MVI M,0CH
0A68 23                 INX H
0A69 229527             SHLD CONTPG
0A6C C38609             JMP STARTC
0A6F CD7D0B  HALT:      CALL KYPAS1   ;PROGRAM GENERATES 8085 CODES
0A72 2A9527             LHLD CONTPG   ;FOR'HLT' INSTRUCTION
0A75 3676               MVI M,76H
0A77 76                 HLT
0A78 CDA30B  HOUT:      CALL KYPAS3   ;PROGRAM GENERATES 8085 CODES
0A7B CDD70B             CALL CONSOT   ;FOR OUTPUTTING THE CONTENTS OF
0A7E 23                 INX H         ;REG.E TO THE SELECTED OUTPUT
0A82 3609               MVI M,09H
0A84 23                 INX H
0A85 229527             SHLD CONTPG
0A88 C38609             JMP STARTC
0A8B CDA30B  OUTC:      CALL KYPAS3   ;O/P THE COMPLEMENT OF
0A8E 2A9527             LHLD CONTPG   ;THE CONTENT OF REG.E
0A91 367B               MVI M,7BH     ;TO THE SELECTED O/P
0A93 23                 INX H
0A94 362F               MVI M,2FH
0A96 23                 INX H
0A97 36E6               MVI M,0E6H
0A99 23                 INX H
0A9A 3601               MVI M,01H
0A9C 23                 INX H
0A90 365F               MVI M,5FH
0A9F 23                 INX H
0AA0 229527             SHLD CONTPG
0AA3 CDD70B             CALL CONSOT
0AA6 23                 INX H
0AA7 36D3               MVI M,0D3H
0AA9 23                 INX H
0AAA 3609               MVI M,09H
0AAC 23                 INX H
0AAD 229527             SHLD CONTPG
0AB0 C38609             JMP STARTC
0AB3 2A9727  STO:       LHLD CODEPG   ;PROGRAM TO STORE THE CONTENTS
0AB6 23                 INX H         ;OF REG.E IN A TEMPORARY
0AB7 7E                 MOV A,M       ;LOCATION 2780H
0AB8 FE1C               CPI 1CH       ;THE 'STO' KEY MUST BE
FOLLOWED
0ABA C2010C             JNZ ERROR     ;BY 'TEMP' KEY,OTHERWISE
```

```
0ABD 23                 INX H           ;ERROR MESSAGE IS DISPLAYED
0ABE 229727             SHLD CODEPG
0AC1 2A9527             LHLD CONTPG
0AC4 36C2               MVI M,0CDH
0AC6 23                 INX H
0AC7 363B               MVI M,3BH
0AC9 23                 INX H
0ACA 360C               MVI M,0CH
0ACC 23                 INX H
0ACD 3621               MVI M,21H
0ACF 23                 INX H
0AD0 3680               MVI M,80H
0AD2 23                 INX H
0AD3 3627               MVI M,27H
0AD5 23                 INX H
0AD6 3673               MVI M,73H
0AD8 23                 INX H
0AD9 229527             SHLD CONTPG
0ADC C38609             JMP STARTC
0ADF 2B      COTEMP:    DCX H           ;IF INSTRUCTION KEY IS
0AE0 7E                 MOV A,M         ;FOLLOWED BY 'TEMP' KEY
0AE1 FE11               CPI 11H         ;PROGRAM BRANCHES TO
0AE3 DA010C             JC ERROR        ;CORRESPONDING 'TEMP' ROUTINES
0AE6 FE17               CPI 17H
0AE8 D2010C             JNC ERROR
0AEB E60F               ANI 0FH
0AED D601               SUI 01H
0AEF 0600               MVI B,00H
0AF1 87                 ADD A
0AF2 4F                 MOV C,A
0AF3 21FC0A             LXI H,TEMTAB
0AF6 09                 DAD B
0AF7 7E                 MOV A,M
0AF8 23                 INX H
0AF9 66                 MOV H,M
0AFA 6F                 MOV L,A
0AFB E9                 PCHL
0AFC 080B    TEMTAB:    DW LDTEM        ;TEMP INSTRUCTION TABLE
0AFE 150B               DW LDCTEM
0B00 2B0B               DW ANDTEM
0B02 3B0B               DW ANDCTM
0B04 540B               DW ORTEM
0B06 640B               DW ORCTEM
0B08 CDEC0B  LDTEM:     CALL TEMRCL     ;ROGRAM GENERATES 8085 CODES
0B0B 23                 INX H           ;LOAD CONTENTS OF
0B0C 365F               MVI M,5FH       ;TEMP LOC. INTO REG.E
0B0E 23                 INX H
```

```
0B0F 229F27            SHLD CONTPG
0B12 C38609            JMP STARTC
0B15 CDEC0B   LDCTEM:  CALL TEMRCL    ;LOAD COMP. OF
0B18 23                INX H          ;CONT. OF TEMP. LOC.
0B19 362F              MVI M,2FH      ;INTO REG.E
0B1B 23                INX H
0B1C 36E6              MVI M,0E6H
0B1E 23                INX H
0B1F 3601              MVI M,01H
0B21 23                INX H
0B22 365F              MVI M,5FH
0B24 23                INX H
0B25 229527            SHLD CONTPG
0B28 C38609            JMP STARTC
0B2B CDEC0B   ANDTEM:  CALL TEMRCL    ;'AND' CONT. OF TEMP.
0B2E 23                INX H          ;LOC. WITH THE CONTENTS
0B2F 36A3              MVI M,0A3H     ;OF REG & STORE THE
0B31 23                INX H          ;RESULT IN REG.E
0B32 365F              MVI M,5FH
0B34 23                INX H
0B35 229527            SHLD CONTPG
0B38 C38609            JMP STARTC
0B3B CDEC0B   ANDCTM:  CALL TEMRCL    ;'AND' THE COMP.
0B3E 23                INX H          ;OF THE CONT. OF THE
0B3F 362F              MVI M,2FH      ;TEMP. LOC. WITH THE
0B41 23                INX H          ;CONT. OF REG.E AND
0B42 36E6              MVI M,0E6H     ;STORE THE RESULT IN
0B44 23                INX H          ;REG.E
0B45 3601              MVI M,01H
0B47 23                INX H
0B48 36A3              MVI M,0A3H
0B4A 23                INX H
0B4B 3657              MVI M,5FH
0B4D 23                INX H
0B4E 229527            SHLD CONTPG
0B51 C38609            JMP STARTC
0B54 CDEC0B   ORTEM:   CALL TEMRCL    ;'OR' THE CONT. OF
0B57 23                INX H          ;THE TEMP. LOC. WITH
0B58 36B3              MVI M,0B3H     ;THE CONT OF REG.E AND
0B5A 23                INX H          ;STORE THE RESULT IN
0B5B 3657              MVI M,5FH      ;REG.E
0B5D 23                INX H
0B5E 229527            SHLD CONTPG
0B61 C38609            JMP STARTC
0B64 CDEC0B   ORCTEM:  CALL TEMRCL    ;'OR' THE COMP.
0B67 23                INX H          ;OF THE CONT. OF
0B68 362F              MVI M,2FH      ;THE TEMP. LOC. WITH
```

```
0B6A 23                 INX H           ;THE CONT OF REG.E AND
0B6B 36E6               MVI M,0E6H      ;STORE THE RESULT IN
0B6D 23                 INX             ;REG.E
0B6E 3601               MVI M,01H
0B71 36B3               MVI M,0B3H
0B73 23                 INX H
0B74 36B3               MVI M,5FH
0B76 23                 INX H
0B77 229527             SHLD CONTPG
0B7A C38609             JMP STARTC

                        ;CONVERSION PROGRAM SUBROUTINES

0B7D 2A9727 KYPAS1:     LHLD CODEPG     ;THIS SUBROU.CHECKS
0B80 23                 INX H           ;WHETHER THE KEY FOLLOWING
0B81 7E                 MOV A,M         ;THIS KEY IS THE '0'
0B82 FE00               CPI 00H         ;DIGIT KEY,OTHERWISE
0B84 C2010C             JNZ ERROR       ;ERROR MESSAGE IS
0B87 C9                 RET             ;DISPLAYED
0B88 2A9727 KYPAS2:     LHLD CODEPG     ;THIS SUBROU.CHECKS
0B8B 7E                 MOV A,M         ;WHETHER THE KEY PRESSED
0B8C FE1C               CPI 1CH         ;IS AN INSTRUCTION KEY
0B8E D2010C             JNC ERROR       ;AND THE KEY FOLLOWING
0B91 23                 INX H           ;THIS KEY MAY BE THE TEMP
0B92 7E                 MOV A,M         ; KEY OR ONE OF THE DIGIT
0B93 23                 INX H           ;ERROR MESSAGE IS DISPLAYED
0B97 2B                 DCX H
0B98 FE1C               CPI 1CH
0B9A CADF0              JZ COTEMP
0B9D FE08               CPI 08H
0B9F D2010              JNC ERROR
0B82 C9                 RET
0BA3 2A9727 KYPAS3:     LHLD CODEPG     ;THIS SUBROU.CHECKS
0BA6 7E                 MOV A,M         ;WHETHER THE KEY PRESSED
0BA7 FE1C               CPI 1CH         ;IS AN INSTRUCTION KEY
0BA9 D2010C             JNC ERROR       ;AND THE KEY FOLLOWING
0BAC 23                 INX H           ;THIS KEY MAY BE ONE
0BAD 7E                 MOV A,M         ;9 THR' F,OTHERWISE
0BAE 23                 INX H           ;ERROR MESSAGE IS DISPLAYED
0BAF 229727             SHLD CODEPG
0BB2 FE1C               CPI 1CH
0BB4 CA010C             JZ ERROR
0BB7 D608               SUI 08H
0BB9 DA010C             JC ERROR
0BBC FE08               CPI 08H
0BBE D2010C             JNC ERROR
0BC1 C9                 RET
```

```
0BC2 2A9527 CONSIN:   LHLD CONTPG   ;THIS SUBROU. CALLS
0BC5 36CD             MVI M,0CDH    ;SUBROU. ROTCNT
0BC7 23               INX H         ;AND ROTIN
0BC8 363B             MVI M,3BH
0BCA 23               INX H
0BCB 360C             MVI M,0CH
0BCD 23               INX H
0BCE 36CD             MVI M,0CDH
0BD0 23               INX H
0BD1 361C             MVI M,1CH
0BD3 23               INX H
0BD4 360C             MVI M,0CH
0BD6 C9               RET
0BD7 2A9527 CONSOT:   LHLD CONTPG   ;THIS SUBROU. CALLS
0BDA 36CD             MVI M,0CDH    ;SUBROU. ROTCNT
0BDC 23               INX H         ;AND ROTOUT
0BDD 363B             MVI M,3BH
0BDF 23               INX H
0BE0 360C             MVI M,0CH
0BE2 23               INX H
0BE3 36CD             MVI M,0CDH
0BE5 23               INX H
0BE6 362B             MVI M,2BH
0BEB 23               INX H
0BE9 360C             MVI M,0CH
0BEB C9               RET
0BEC 2A9527 TEMRCL:   LHLD CONTPG   ;THIS SUBROU. CALLS
0BEF 36CD             MVI M,0CDH    ;SUBROU. ROTCNT
0BF1 23               INX H         ;AND LOADS THE CONTENTS
0BF2 363B             MVI M,3BH     ;OF THE TEMP. LOC. INTO
0BF4 23               INX H ;ACC.
0BF5 360C             MVI M,0CH
0BF7 23               INX H
0BF8 363A             MVI M,3AH
0BFA 23               INX H
0BFB 3680             MVI M,80H
0BFD 23               INX H
0BFE 3627             MVI M,27H
0C00 C9               RET
0C01 3600   ERROR:    MVI M,00H     ;THIS SUBROU. FLICKERS
0C03 47               MOV B,A       ;ERROR MESSAGE 'ERUR'
0C04 217C27 FLIKER:   LXI H,277CH   ;IN THE ADDRESS FIELD
0C07 CDB505           CALL OUTPUT
0C0A 1100A0           LXI D,0A000H
0C0D CDA003           CALL DLAY
0C10 CD2B03           CALL CLEAR
0C13 110050           LXI D,5000H
```

```
0C16 CDA003            CALL DLAY
0C19 C3040C            JMP FLIKER

                       ;CONVERTED PROGRAM SUBROUTINES

0C1C DB08     ROTIN:   IN 08H          ;THIS SUBROU. SHIFTS
0C1E 04                INR B           ;THE STATUS OF THE
0C1F 05                DCR B           ;SELECTED I/P TO THE
0C20 CA280C            JZ DOLL         ;'O' BIT POSITION
0C23 0F       STAR:    RRC
0C24 05                DCR B
0C25 C2230C            JNZ STAR
0C28 E601     DOLL:    ANI 01H
0C2A C9                RET
0C2B 78       ROTOUT:  MOV A,B         ;THIS SUBROU. SHIFTS
0C2C D608              SUI 08H         ;THE CONT. OF REG.E
0C2E 47                MOV B,A         ;TO THE SELECTED O/P
0C2F 7B                MOV A,E         ;BIT POSITION
0C30 04                INR B
0C31 05                DCR B
0C32 CA3A0C            JZ YEN
0C35 07       DINAR:   RLC
0C36 05                DCR B
0C37 C2350C            JNZ DINAR
0C3A C9       YEN:     RET
0C3B 2A9727   ROTCNT:  LHLD CODEPG     ;THIS SUBROU. STORES
0C3E 23                INX H           ;THE ADD. OF THE
0C3F 46                MOV B,M         ;SELECTED I/P OR
0C40 23                INX H           ;O/P IN REG.E
0C41 229727            SHLD CODEPG
0C44 C9                RET
0C45 D5       WAIT:    PUSH D          ;THIS SUBROU. GENERATES
0C46 0E04     TAKA:    MVI C,04H       ;THE DELAY OF ONE SECOND
0C48 16D0     SHILNG:  MVI D,0D0H
0C4A 1EFF     PAISE:   MVI E,0FFH
0C4C 1D       CENTS:   DCR E
0C4D C24C0C            JNZ CENTS
0C50 15                DCR D
0C51 C24A0C            JNZ PAISE
0C54 0D                DCR C
0C55 C2480C            JNZ SHILNG
0C58 05                DCR B
0C59 C2460C            JNZ TAKA
0C5C D1                POP D
0C5D C9                RET
```

BIBLIOGRAPHY

1. PROGRAMMABLE LOGIC CONTROLLERS

   - MARCO COSTANZO

2. UNDERSTANDING AND USING PROGRAMMABLE CONTROLLERS

   - THOMAS E. KISSEL

3. PROGRAMMABLE LOGIC CONTROLLERS (Principles and applications)

   - JOB W.WEBB

   - RONALD A. REIS

4. INTRODUCTION TO PROGRAMMABLE LOGIC CONTROLLERS (The Mitsubishi-FX)

   - JOHN RIDLEY

5. GEAR HOBBING MACHINE MANUAL (WAHLI-91)