Electronics 6th Sem PLC

* Introduction to PLCs.

*** What is PLC -** A programmable logic controller is a ruggedized computer used for industrial automation. These controllers can automate a specific process, machine function or even an entire production line. PLC adapted for the control of manufacturing processes. such as assembly lines or robotic devices or any activity that requires high reliability, ease of programming and process fault diagosis.

* PLC can range from small modular devices with tens of inputs and outputs in a housing interval with the processor. to large rack-mounted modular devices with a count of thousand of I/O and which are often networked to other PLC an SCADA systems.

* They can be designed for many arrangements of digital and analog I/O extended temperature ranges, immunity to electrical noise. and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed up or non-volatile memory.

* PLCs were first developed in the automobile manufacturing industry to provide flexible, rugged and easily programmable controllers to replace hard-wired relay logic systems. Since then they have been widely adopted as high-reliability automation controllers. suitable for harsh environments.

A PLC is an example of a "hard" real time system since o/p result must be produced in response to input conditions. within a limited time, otherwise unintended operation will result.

+ Limitation of Relays -

* The Relay cannot switched on and off at high speeds because they have slow response and the switch contacts will rapidly wear out due to the sparking.
* Their coils need a fairly high current to energised, which means some micro-electronic circuits can't drive them directly without additional circuitry.
* The back-emf created when the relay coil switch off can damage the components that are driving the coil. to avoid this, a diode can be placed accross the relay coil.
* Their parts can wear out as the switch contacts becomedirty high voltages and current causes sparks between the contacts

+ Advantages of PLC over Relay -

* PLCs are programmable
* Many control relay can be replaced by software.
* Less hardware failure.
* Easier to change in software.
* Special function such as counter, timer, delay provi
* More Reliabile.
* No task limit
* Cost effective control system

Block diagram of PLC –

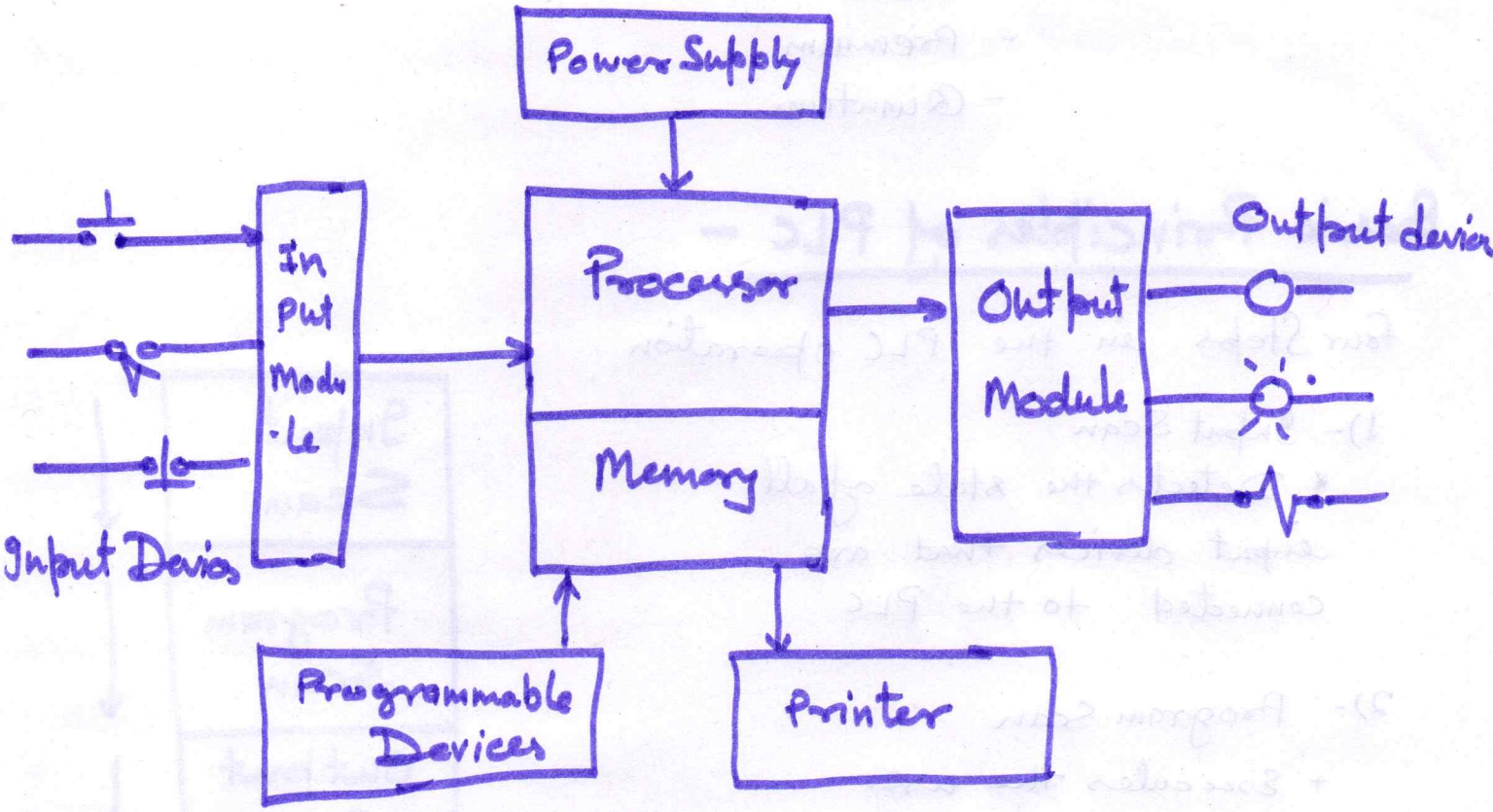


Fig. Block diagram of PLC

Components –

1)- Central processing Unit (CPU) – 8 bit, 16, 32 bit

2)- Input output modules (analog 0-10V, 4-20mA, digital 24Vdc 115V ac, 230V ac)

3)- Remote input output modules (Attached with PLC to computer)

4)- Power supply

5)- Bus system

PLC Manufacturers –

1)- Siemens – S5 - 110 U, 115 U, 135 U
– S7 - 200, 300, 400

2)- Allen Bardley – Micrologix 1000, 1200, 1500
– SLC 5/01, 5/02, 5/03
– PLC 5/10, 5/25, 5/40

3- Modicon - Nano
- Micro
- Premium
- Quantum.

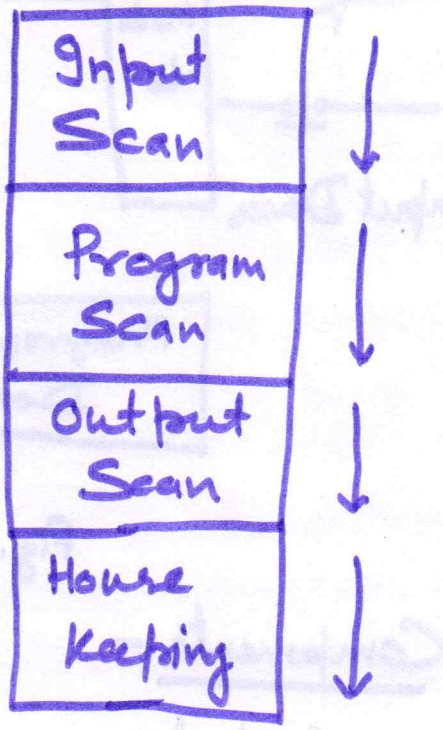
Basic Principles of PLC -

four Steps in the PLC operation

1)- Input Scan
* Detects the state of all input devices that are connected to the PLC

2)- Program Scan
* Executes the user created program logic

3)- Output Scan
* Energises or deenergised all output devices that are connected to the PLC.



4)- Housekeeping.
* This step includes communication with programming terminals, internal, diagnostics etc.

Programming Language - While ladder logic is the most commonly used PLC programming language. it is not the only one.

Ladder Diagram - (LD) - Traditional ladder logic is graphical programming Language - Initially programmed with simple contacts that simulated the opening and closing the relays, ladder logic programming has been expanded to include such function

Counter, timers, shift registers and math operations.

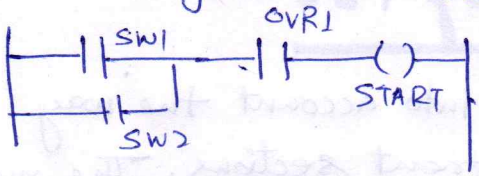


fig. Ladder diagram.

* Functional Block Diagram (FBD) - A graphical language for detecting signal and data flows through re-usable function blocks. FBD is very useful for expressing the inter connectio of control system algorithms and logic.

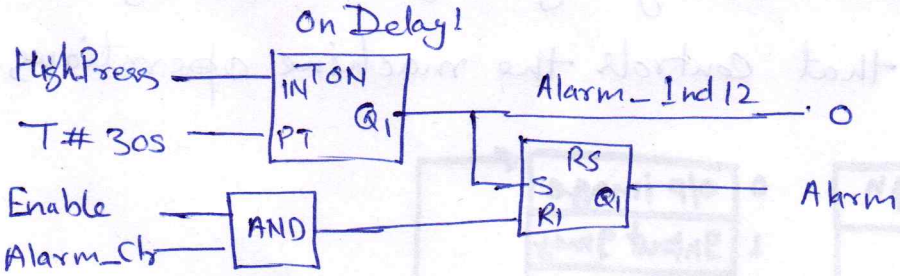


fig. FBD

* Structured Text (ST) - A high level text language that encourages structured programming. It has a language structure (syntax) tha strongly resembles PASCAL and supports a wide range of standard function and operators.

* Instruction list (IL) - A low level "assembler like" languag that is based on similar instruction set or list language found in a wide range of today's PLCs.

* Sequential Function Chart (SFC) - A method of programming complex control system at a more highly structured level. A SFC program is an overview of the control system in which the basic building blocks are entire program files. Each file program is create using one another type of programming language. The SFC approa coordinates large complicated programming tasks into smaller, more manageable tasks.

Memory Orgnization of PLC –

Memory orgnization takes into account the way a PLC divides the available memory into different sections. The memory space can be divided into two broad categories. ① Program files ② Data files.

Program files –

Program files are the part of the processor memory that stores the user ladder logic program. The program accounts for most of the total memory of a given PLC system. It contains the ladder logic that controls the machine operations.

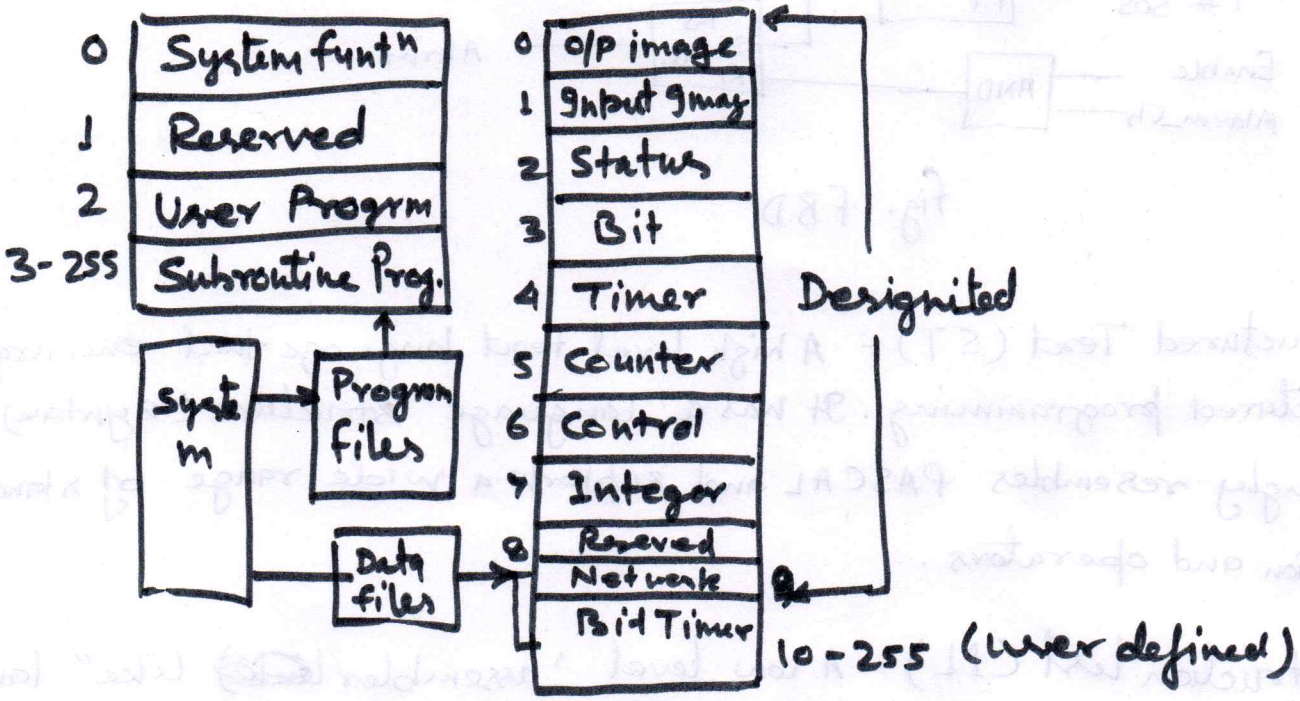


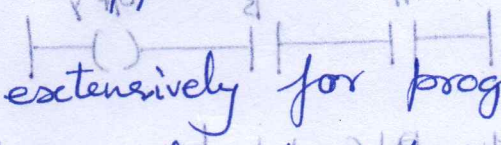
Fig. - Program files memory.

Program files are the areas of processor memory where ladder logic programming is stored.

PLC

Topic - **Ladder programming.**

* Ladder Logic Basics - Ladder logic is a programming language that is used to program a PLC. It is a graphical PLC programming language which express logic operations with symbolic notations using ladder diagrams. much like the rails and rungs of the traditional relay logic circuits. It is used to execute logical sequential, counting, timing and arithmetic tasks in order to carry industrial automation applications.



Ladder logic is used extensively for programming PLCs in a multitude of industrial automation applications.

Some examples are

* Material Handling conveyer system
* Pallet Packing and Strapping.
* Ball Mill lubrication systems
* Logistics Package conveying and Sorting.
* Cement Batching.
* Beverage Bottling and Labelling.
* Hopper and tank level Controlling.
* Air and Liquid flow and pressure ~~flow~~ Control.

* NOT function Ladder Logic.

NOT is basically the opposite state of an event that occurs.

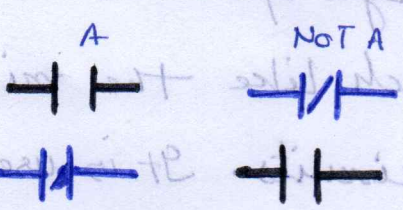
if PLC input A is false the result will be True.

And vise versa when PLC input A is true the result will be false.

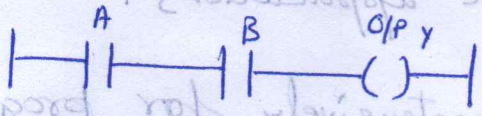
* Not function is reffered as reverse logic.

A	NOT A
FALSE	TRUE
TRUE	FALSE

Ladder logic is

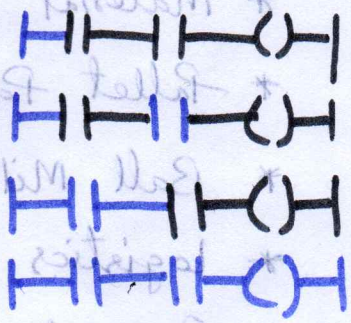


* AND Function - Examine multiple PLC inputs has one output

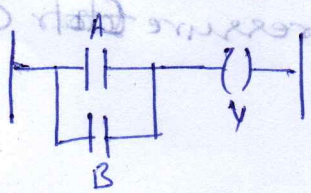


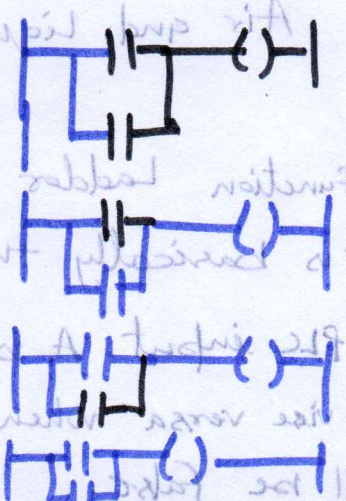
* If all the PLC inputs are true the corresponding result also True.

Inputs		outputs
A	B	y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE



* OR Function - This is equivalent of the parallel connection in an electronic circuit.





Input		Output
A	B	y
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

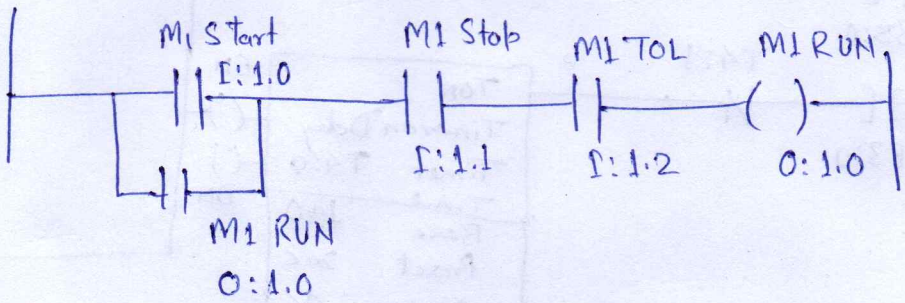
* Motor Control Ladder Diagram -

* Required Inputs.

Name	Address	type	Comment	PLC wiring.
M1 START	I:1.0	BOOL	Motor No 1 Start Push button	Normally open
M1 STOP	I:1.1	BOOL	Motor No 1 Start Push Button	Normally closed
M1 TOL	I:1.2	BOOL	Motor No.1 Thermal Overload	Normally Closed

* Outputs.

M1 RUN	O:1.0	BOOL	Motor No1 RUN	Normally Open.



Ladder Logic for motor control.

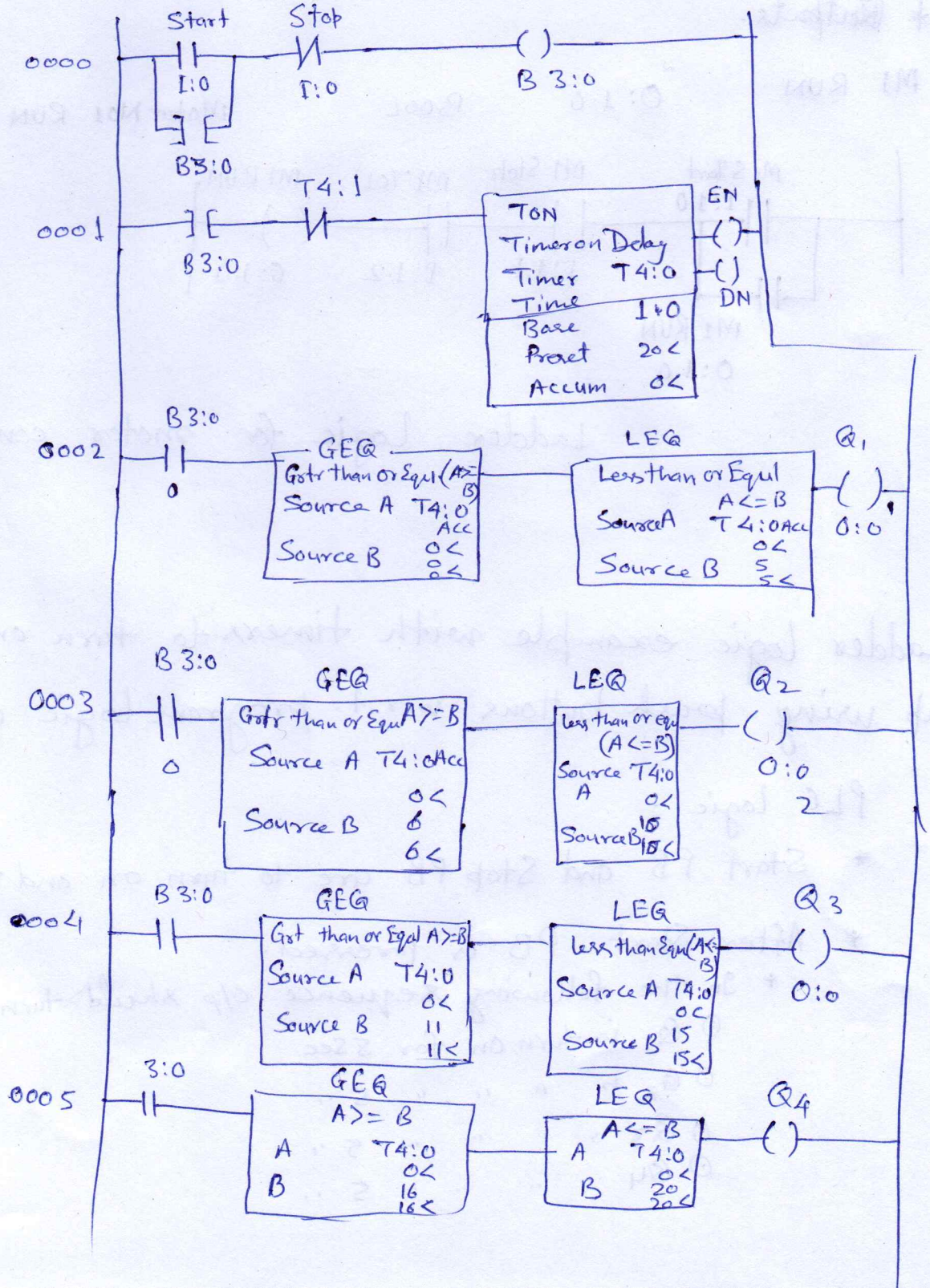
* Ladder logic example with timers to turn on/off the lamp using push buttons w.r.t. program logic conditions.

PLC logic -

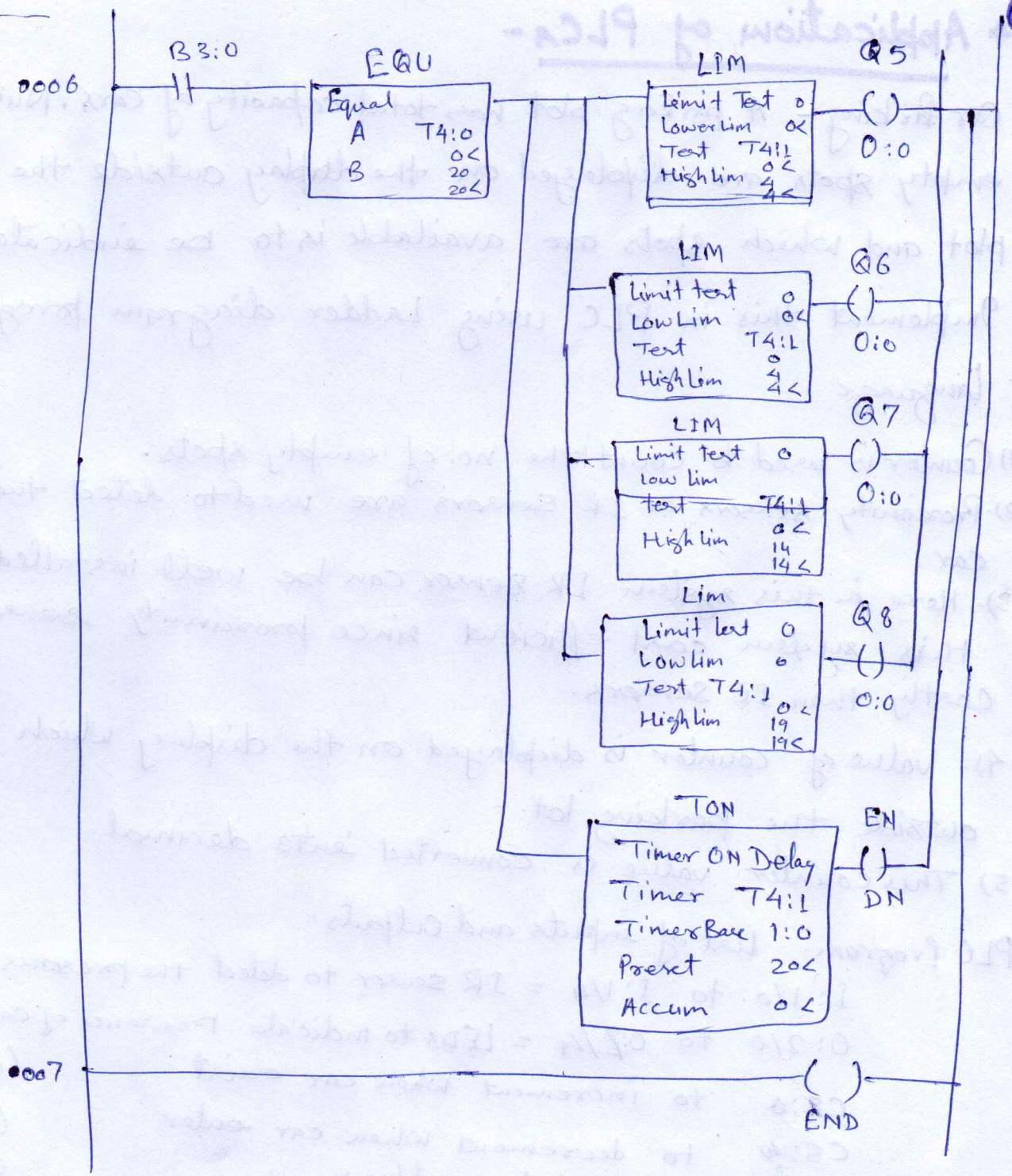
* Start PB and Stop PB are to turn on and turn off lamp.
* After Start PB is pressed.
  * In the following sequence O/p should turn ON/OFF.
    - Q_1 is turn on for 5 Sec
    - Q_2 is " " " 5 "
    - Q_3 " " " " 5 "
    - Q_4 " " " " 5 "

* End of above sequence will trigger.
  - All four o/p (Q5, Q6, Q7, Q8) to turn ON.
  - In the following sequence o/p should turn OFF after its turn ON.
  - After 5 sec Q5 should go OFF.
  - " " " Q6 " " "
  - " " " Q7 " " "
  - " " " Q8 " " "

* Process should continue until stop PB is pressed.



Continuee.——


0006
B3:0
EQU
Equal
A T4:0
B
LIM
Limit Test
Lower Lim
Test T4:1
High Lim
O:0
LIM
Limit test
Low Lim
Test T4:1
High Lim
O:0
LIM
Limit Test
Low Lim
Test T4:1
High Lim
O:0
Lim
Limit test
Low lim
Test T4:1
High Lim
O:0
TON
Timer ON Delay
Timer T4:1
Timer Base 1:0
Preset 20
Accum
EN
DN
0007
END

* Application of PLCs -

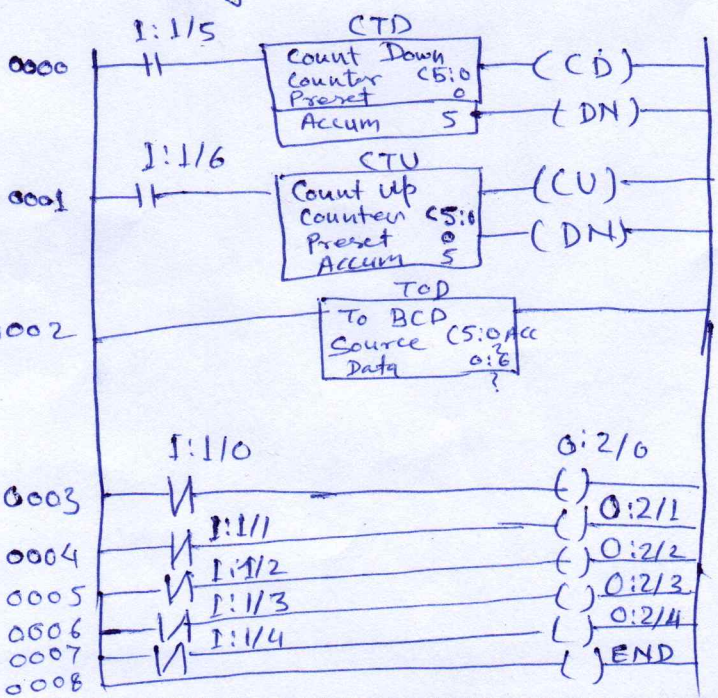
* Car Parking - A parking plot has total capacity of cars. Numbers of empty spots are diplayed on the display outside the parking plot and which spots are available is to be indicated by LEDs. Implement this in PLC using Ladder diagram programming language -

1) Counter is used to count the no. of empty spots.
2) Proximity sensors or IR Sensors are used to detect the presence of car.
3) Here in this system IR sensor can be well installed to make this system cost efficient since proximity sensor are costly than IR sensors.
4) Value of counter is displayed on the display which is mounted outside the parking lot.
5) This counter value is converted into decimal.

PLC Program. List of inputs and outputs.

I:1/0 to I:1/4 = IR sensor to detect the presence of cars (input
O:2/0 to O:2/4 = LEDs to indicate presence of car spots (output
C5:0 to increment when car exist (counter up)
C5:0 to decrement when car enter (counter down)
O:6 to Display address (output)

Ladder diagram.



Data Files – The data files portion of the processor's memory stores input and output status, processor status, the status of various bits and numerical data. All this information is accessed via the ladder logic program. These files are organised by the type of data they contain and may include.

* Output (file 0) – This file stores the state of the O/P terminals for the controller.
* Input (file 1) – This file stores the status of the inputs terminals for the controller.
* Status (file 2) – This file stores the controller operation information and is useful for troubleshooting controller and program operation.
* Bit (file 3) – This file is used for internal relay logic storage.
* Timer (file 4) – This file stores the timer accumulated and preset values and status bits.

Allen Bardley PLC Processor memory organization –
This processor operates in two different data type. integer and floating point. All data types, except the floating-point files are treated as integers or whole numbers. All elements and bit addresses in the output and input data files are numbered actually. Element and bit addresses in all other data files are number decimally.

The PLC stores all data in global data tables and are based on 16 bit operations. Typical addressing format for the PLC-5 controller are as follows –

* The addresses in the output data files and the input data files are potential locations for either input modules or O/P modules mounted in the I/O chassis.

* The Status data file contains information about the processor status.
* The bit data file stores bit status. It frequently serves as storage when using internal outputs, sequencers, bit shift inst^{n} and logical inst^{n}.
* The timer file stores the timer status and timer data.
* The counter file stores the counter status and counter data.
* The control file stores the control element's status and data.
* The integer file stores integer data values with a range from 232,768 through 32,767, in decimal form. single word (16 bit)
* The floating point file can store values in range from 61.17549 44e-38 to 63.40282 37e138. It is 2 word long.
* Data files 9 through 999 may be assigned to different data types as required. When assigned to ~~different~~ certain type, a file is then reserved for that type and cannot be used for any other type. Additional input, output or status files cannot be created.