Programmable Logic Controllers

Programmable Logic Controllers

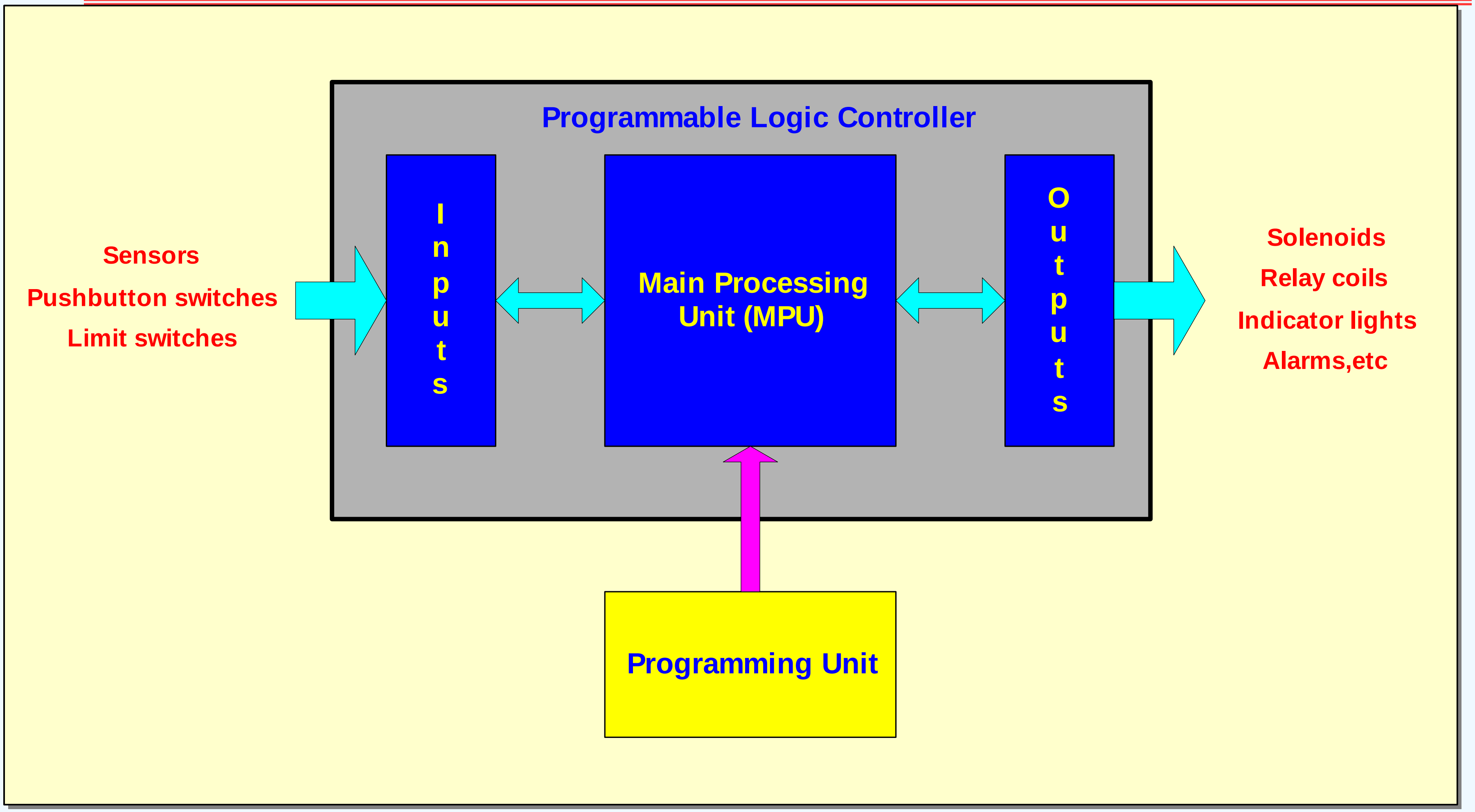
- microcomputer-based controllers
- can be programmed for sequence control purposes
- other powerful features: counting and timing, arithmetic processing, process control, etc.
- provides flexible automation; *reprogrammable.* Ladder diagrams can be programmed into the PLC
- cost-effective for medium- or large-sized applications
- takes up less space; can often replace several hundred relays.

Programmable Logic Controllers

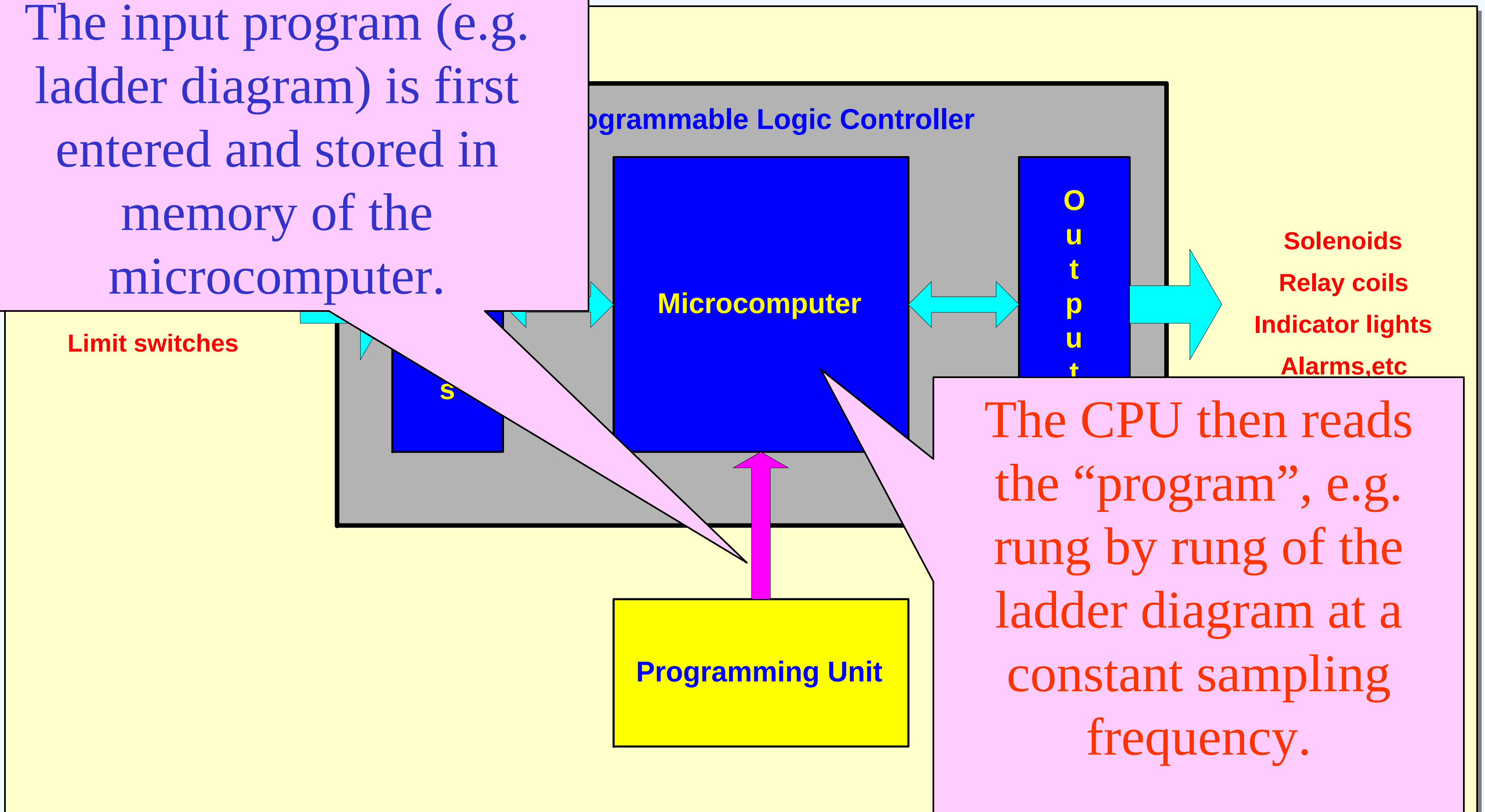
•generally more reliable than relay circuits. Relays have life-cycles of the order of hundreds of thousands while that for the PLC are in millions.

•eliminates the often appreciable cost of wiring a relay panel.

Basic Architecture of PLC



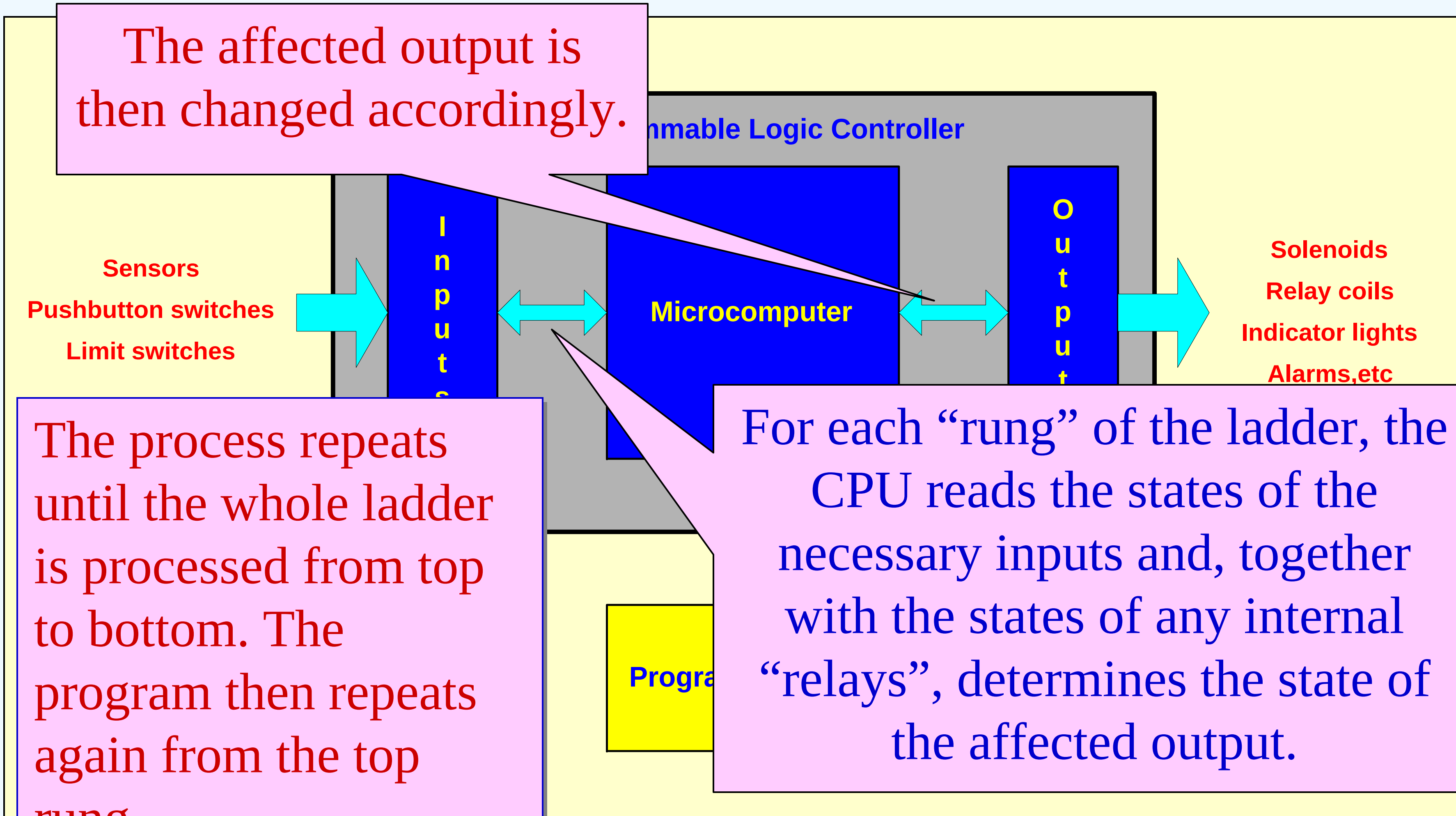
Basic Operation of the PLC



The input program (e.g. ladder diagram) is first entered and stored in memory of the microcomputer.

The CPU then reads the "program", e.g. rung by rung of the ladder diagram at a constant sampling frequency.

Basic Operation of the PLC



The affected output is then changed accordingly.

For each "rung" of the ladder, the CPU reads the states of the necessary inputs and, together with the states of any internal "relays", determines the state of the affected output.

The process repeats until the whole ladder is processed from top to bottom. The program then repeats again from the top rung.

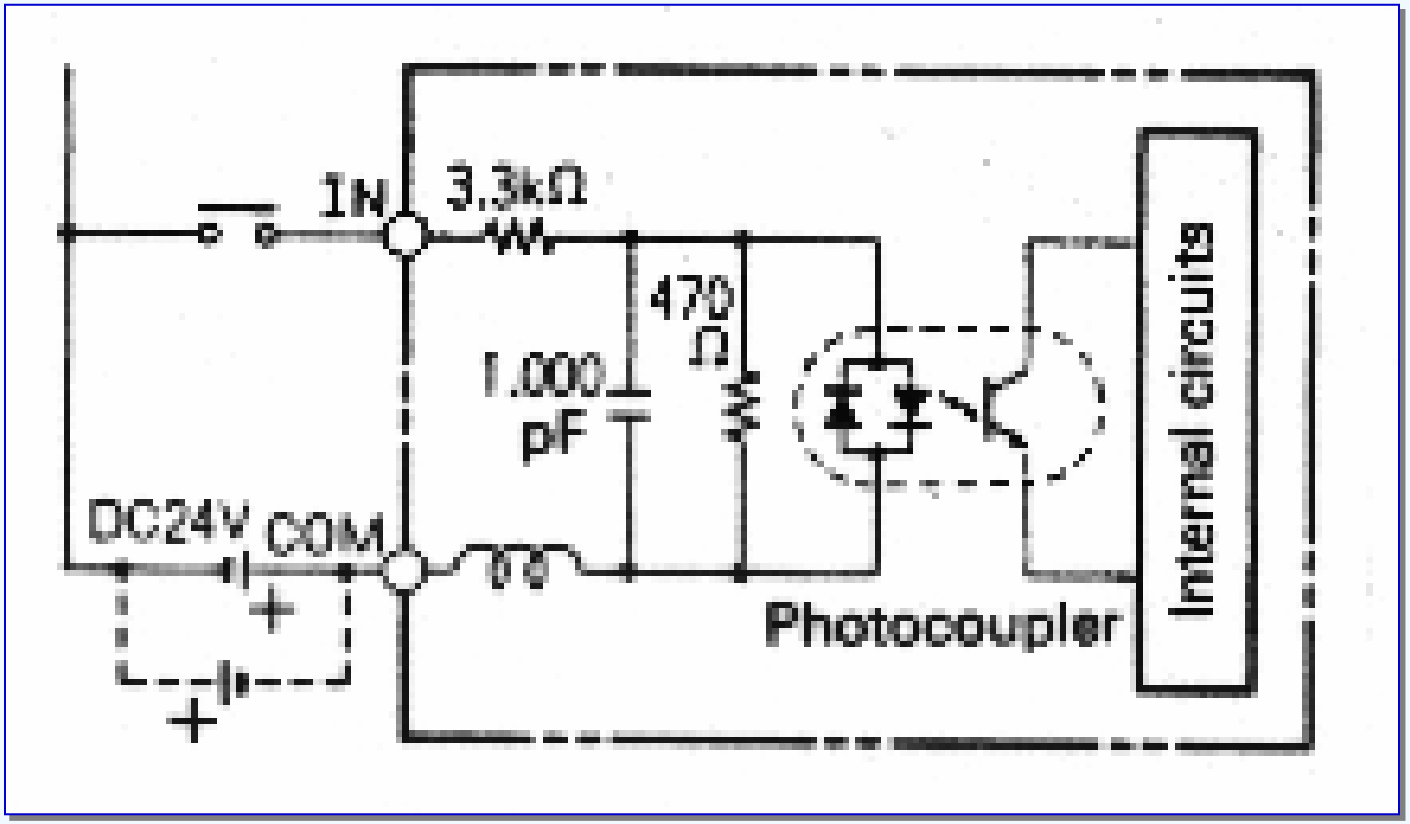
A typical commercial PLC

Typical PLC Specifications

Item	Specification
Step Execution Time	3.6 microsecs max
Program Capacity	1.6K (800 program steps)
Size of Instruction Set	20 basic, 2 step ladder, 35 applied (57 total)
Power Supply	100-240 VAC, 12 VDC depending on model
Maximum Service Supply	200 mA at 24 VDC
Maximum Number of I/O	16 inputs, 14 outputs (depends on model; numbered in octal)
Input Rating	24 VDC or 120 V (85-132 VAC) depending on model
Max. Output Relay Current	2.5 A / point 8A / 4 point common (250 VAC / 30 VDC)
Max. Transistor O/P Current	0.5 A / point, 0.8 A / 4 point common (30 VDC)
Device Table	512 general purpose internal flags (auxiliary relays) 56 special purpose internal flags (auxiliary relays) 64 states (for use in Step Ladder programs) 56 timers (minimum delay 0.01 sec) 15 counters 4 single-phase high speed counters (maximum frequency 7kHz) or; 1 quadrature high speed counter (maximum frequency 2kHz) 32 data registers (including 2 non-volatile) 27 special purpose data registers

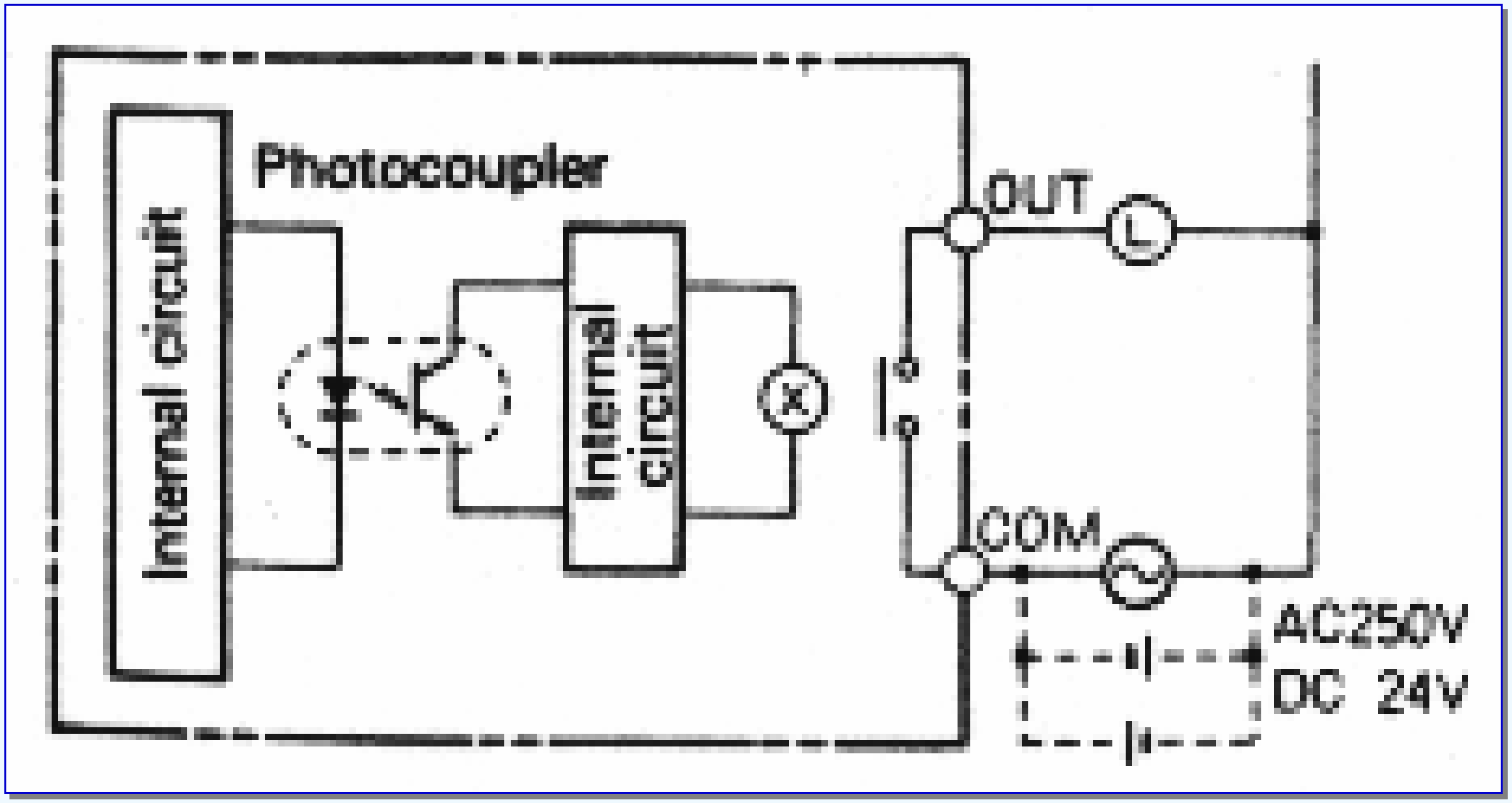
Typical Input Interface Circuit

To detect state of sensors, switches, etc.



Typical Output Interface Circuit

To translate low voltage/current signals to large current, high voltage outputs.



Programming the PLC

Various methods available.

Varies from manufacturer to manufacturer.

Done directly through a programming unit or through a connection to a PC, depending upon the PLC used.

Actual programming is a relatively easy process. The difficult part is deriving the logic, or ladder diagram, required.

Programming the PLC

- **ladder-diagram based**
  - carry-over from the popular ladder diagram approach used in the past and familiar to many automation engineers
  - ladder diagram or Boolean expressions of all logic function first derived and “program” input via graphical means of text editor.
- **Instruction List (mnemonics)**
  - Low level language similar to assembly language codes.
- **Others:** sequential function chart (SFC), function block diagram (FBD),structured text (ST).

Programming the PLC(based on Mitsubishi)

Basic Devices

Xn:

- Reserved for physical input devices, e.g. limit switches, pushbuttons, sensors, connected directly to inputs of PLC.
- Number n available limited depending upon PLC used.
- State of contacts corresponds directly to the physical input connection to the PLC.
- Any number of contacts available for program.

Yn:

- Reserved for physical output device, e.g. relays, solenoids, motors, connected directly to outputs of PLC.
- Number n available is limited depending upon PLC used.
- These are also implemented as “software” relays with many contacts.

Programming the PLC(based on Mitsubishi)

Basic Devices

Tn:

- internal "software" timers used for generating time delays. Number n almost unlimited.

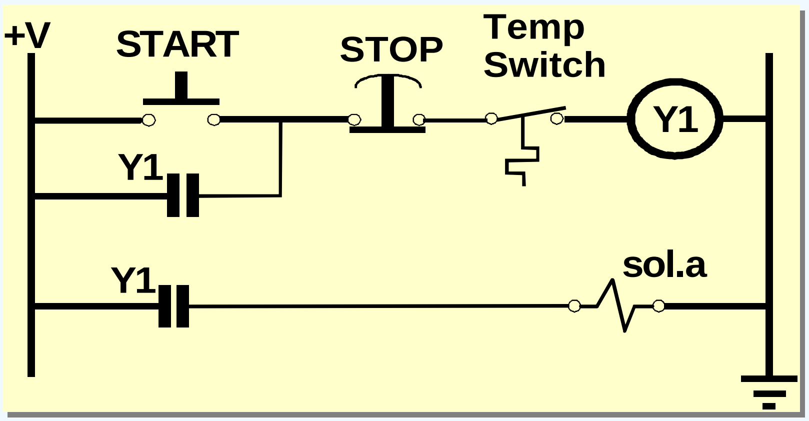
Mn:

- internal auxiliary "software" relays which comprises a coil and contacts. Number n almost unlimited.

Cn:

- internal "software" counters for counting events. Number n almost unlimited.

Preparing for the Program

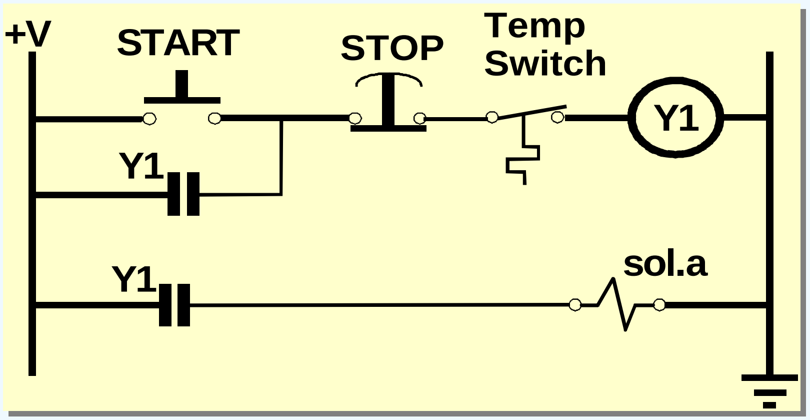


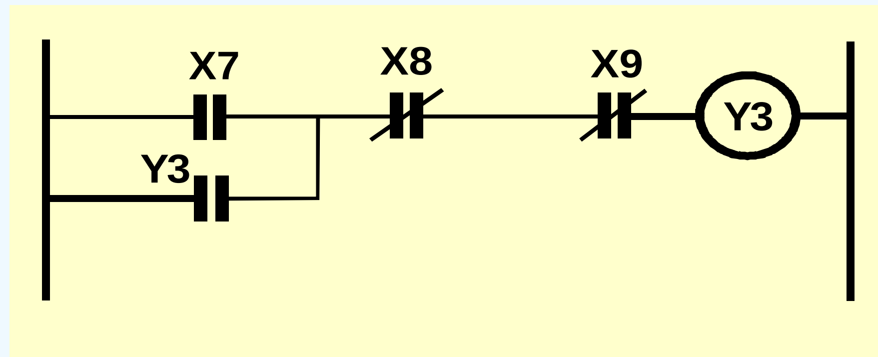
The original ladder diagram

Input Module	Output Module	Connected to
X7		START
X8		STOP
X9		Temperature Switch
	Y3	Sol a

Input-Output assignments

Preparing for the Program

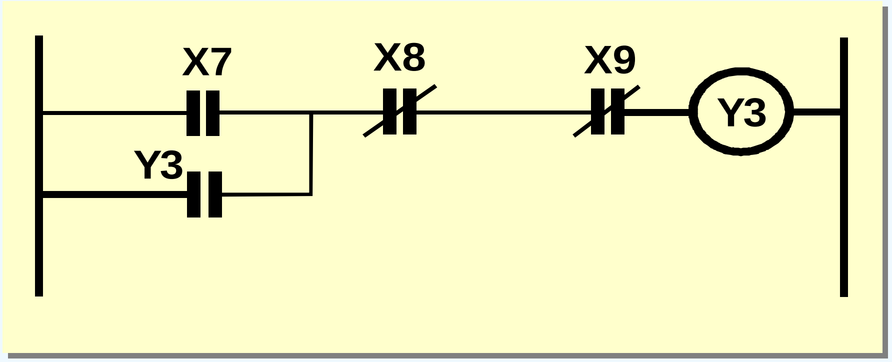




Input Module	Output Module	Connected to
X7		START
X8		STOP
X9		Temperature Switch
	Y3	Sol a

Equivalent PLC ladder diagram

A Sample Programming Language

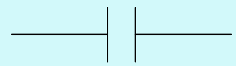
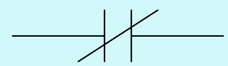
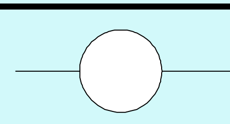


Ladder Diagram

```
0   LD   X7
1   OR   Y3
2   ANI  X8
3   ANI  X9
4   OUT  Y3
5   END
```

Program

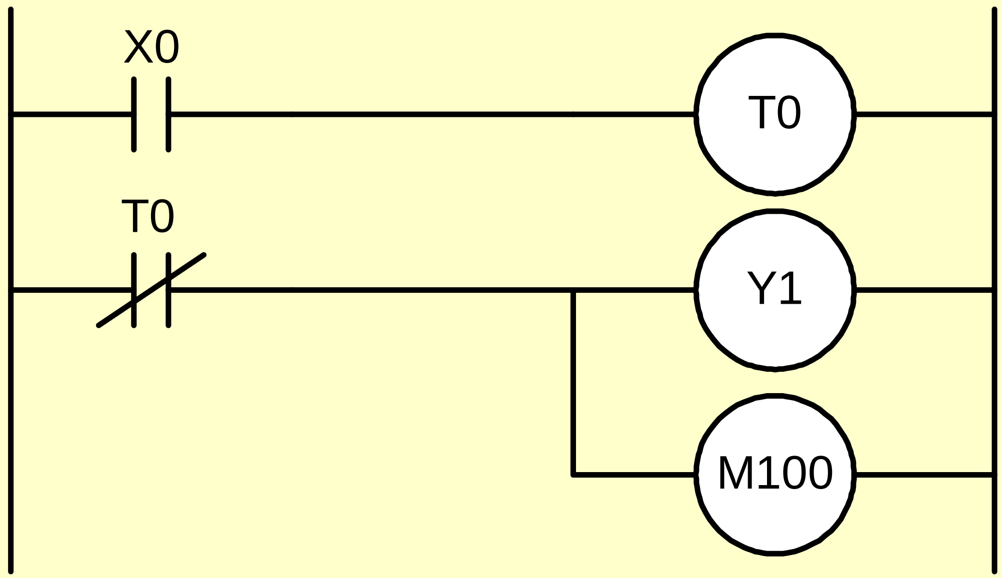
Mnemonic Instructions (based on Mitsubishi)

Mnemonic	Function	Figure	Devices	Example
LD (Load)	Initial logical operation – NO contacts		X,Y,T,M,C	LD X3
LDI (Load Inverse)	Initial logical operation – NC contacts		X,Y,T,M,C	LDI X3
OUT (Out)	Final logical operation - connects to right rail		Y,T,M,C	OUT Y3

The LD and LDI instruction initiates a new logical block. Y,T,M,C would be contacts associated with the respective devices.

OUT connects output device to right hand rail or bus bar. Cannot be used with X input devices. Multiple parallel connections allowed.

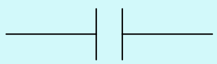
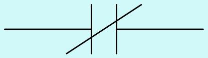
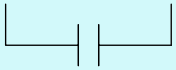
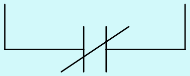
Mnemonic Instructions (LD, LDI, OUT)



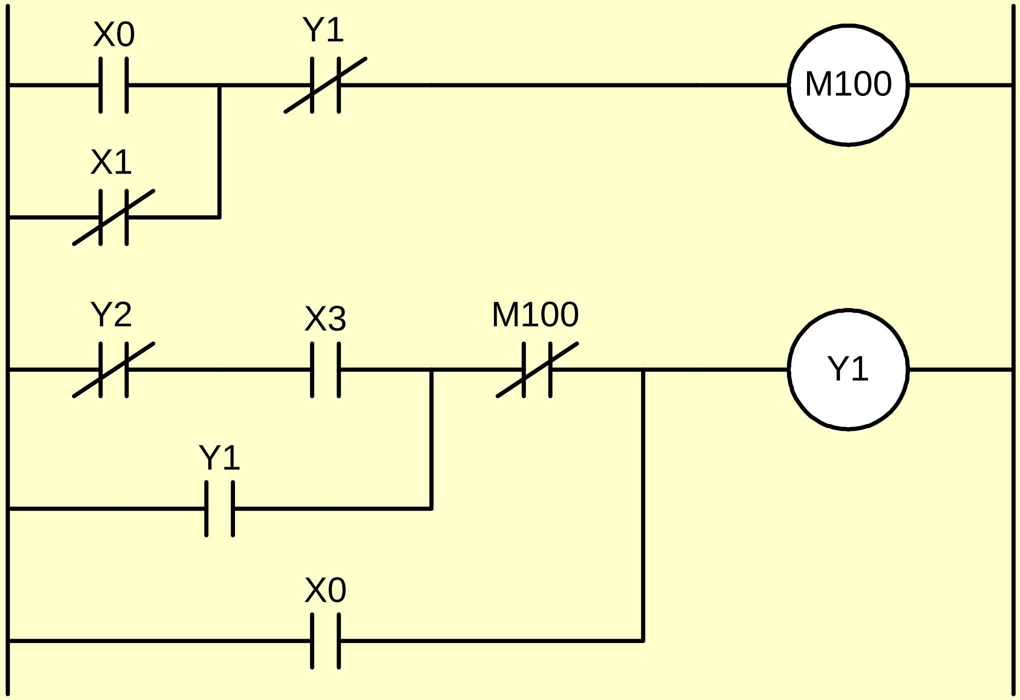

```
LD  X0
OUT T0 K19
LDI T0
OUT Y1
OUT M100
END
```

19x100msec

Mnemonic Instructions (AND,ANI,OR,ORI)

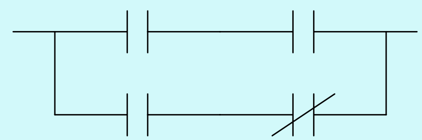
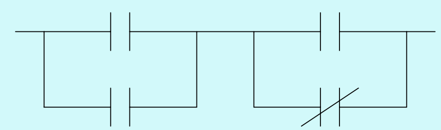
Mnemonic	Function	Figure	Devices	Example
AND (AND)	Serial connection – NO contacts		X,Y,T,M,C	AND X 3
ANI (AND Inverse)	Serial connection – NC contacts		X,Y,T,M,C	ANI Y 3
OR (OR)	Parallel connection – NO contacts		X,Y,T,M,C	OR M 3
ORI (OR Inverse)	Parallel connection – NC contacts		X,Y,T,M,C	ORI X 3

Mnemonic Instructions (AND,ANI,OR,ORI)

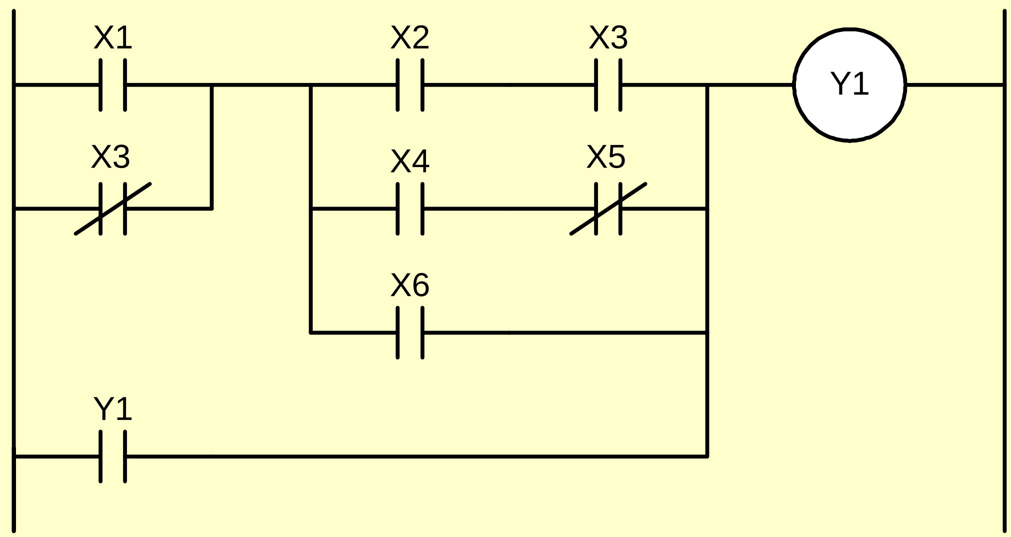



```
LD  X0
ORI X1
ANI Y1
OUT M100
LDI Y2
AND X3
OR  Y1
ANI M100
OR  X0
OUT Y1
END
```

Mnemonic Instructions (ORB,ANB)

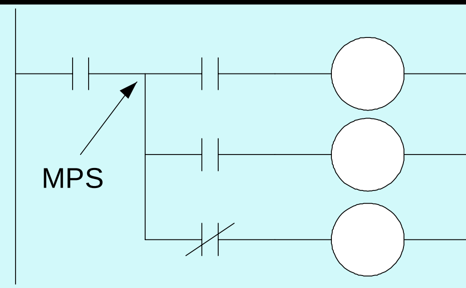
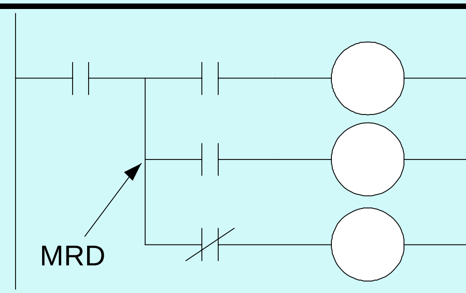
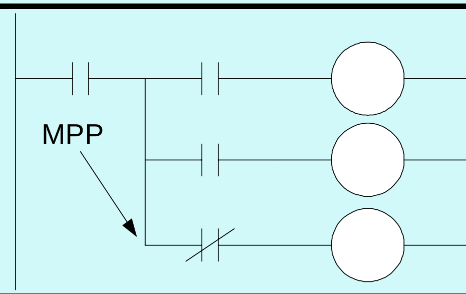
Mnemonic	Function	Figure	Example
ORB (OR block)	Parallel connection of multiple serial circuits		ORB
ANB (AND block)	Serial connection of multiple parallel circuits		ANB

Mnemonic Instructions (ORB,ANB)

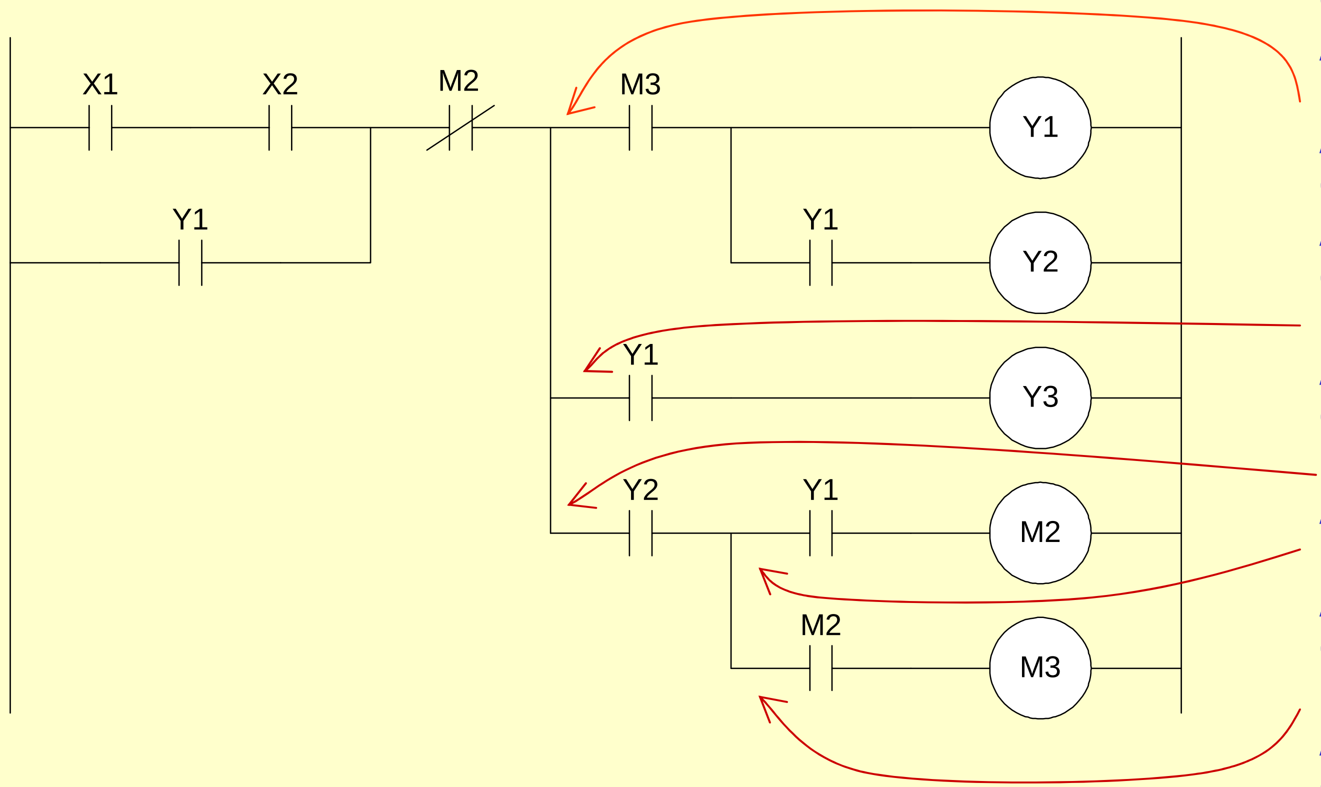



```
LD   X1
ORI  X3
LD   X2
AND  X3
LD   X4
ANI  X5
ORB
OR   X6
ANB
OR   Y7
OUT  Y1
END
```

Mnemonic Instructions (MPS,MRD,MPP)

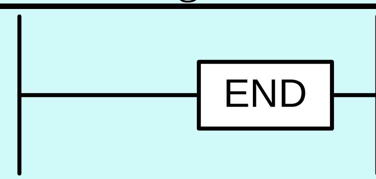
Mnemonic	Function	Figure	Example
MPS (Point Store)	Stores current result		MPS
MRD (Read)	Reads current result		MRD
MPP (Pop)	Pops (reads and removes) currently stored result		MPP

Mnemonic Instructions (MPS,MRD,MPP)



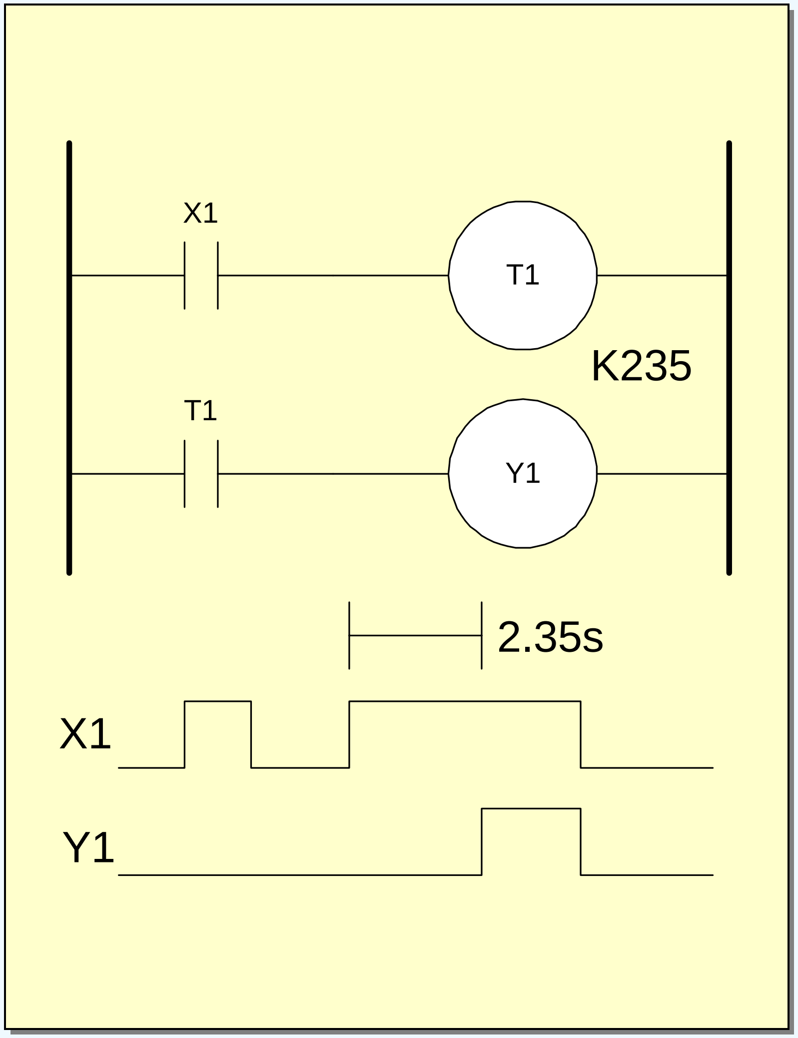

```
Ld  X1
AND X2
OR  Y1
ANI M2
MPS
AND M3
OUT Y1
AND Y1
OUT Y2
MRD
AND Y1
OUT Y3
MPP
AND Y2
MPS
AND Y1
OUT M2
MPP
AND M2
OUT M3
END
```

Mnemonic Instructions (END)

Mnemonic	Function	Figure
END	Forces current program scan to end.	

- **END forces program to end current scan and restart a new scan.**
- **Useful for debugging purposes as instructions after END are ignored.**

Timers

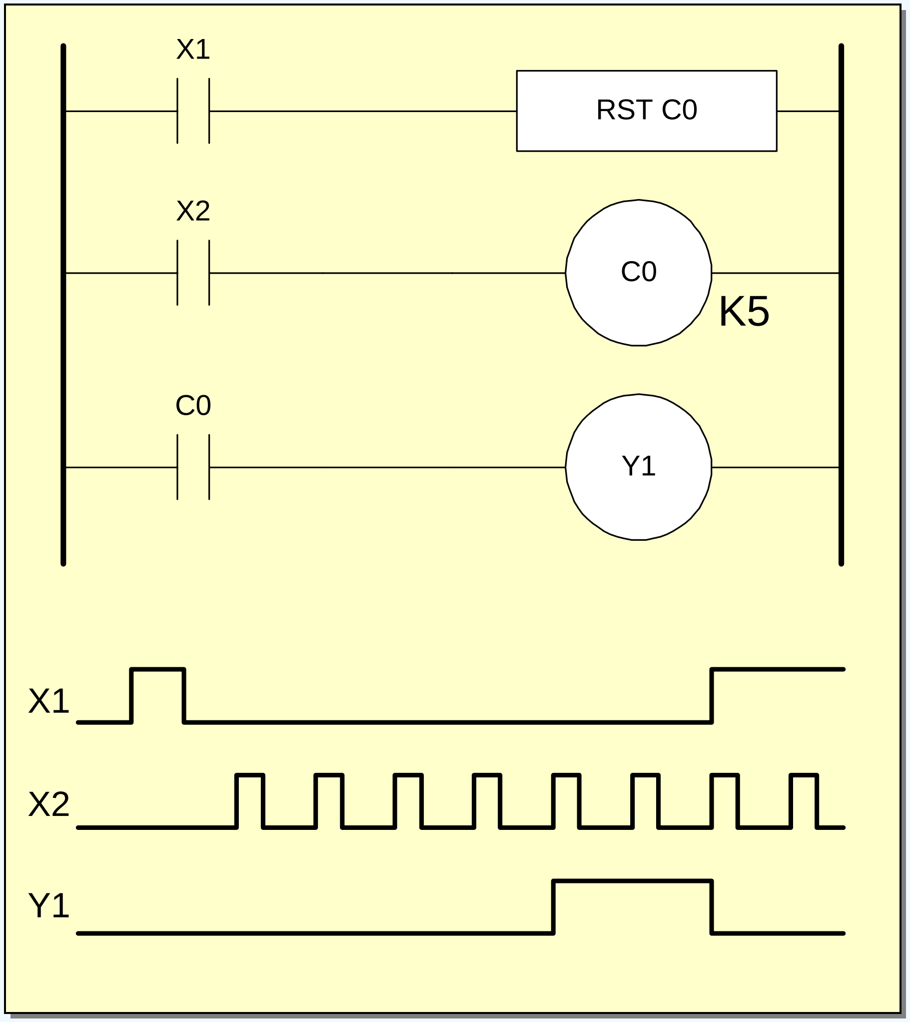


- "Software" timers are normally available in PLCs. In the Mitsubishi PLC, there are 1msec, 10msec and 100msec timers. These have timer "Coils" and "Contacts".

```
Ld   X1
OUT T100 K235
Ld   T1
OUT Y1
```

- If T1 is a 10msec timer, then relay Y1 will turn on after 2.35 sec after X1 closes and remains closed. At any time X1 opens, T1 resets.

Counters




```
Ld  X1
RST C0
Ld X2
OUT C0 K5
Ld  C0
OUT Y1
```

Closure of X1 resets the counter C0. C0 counts up each time its coil is turned ON by X2. Its output contacts are activated when its coil is turned ON for the fifth time. Thereafter its count value does not change and its outputs remain ON until it is reset to zero X1 closing.

Program Scan

Program Scan:

A single processing of the loaded program from start to END. The process is continuous and once one scan ends, a new one is started.

Scan Time:

Time period for one scan, dependent upon program length and complexity.

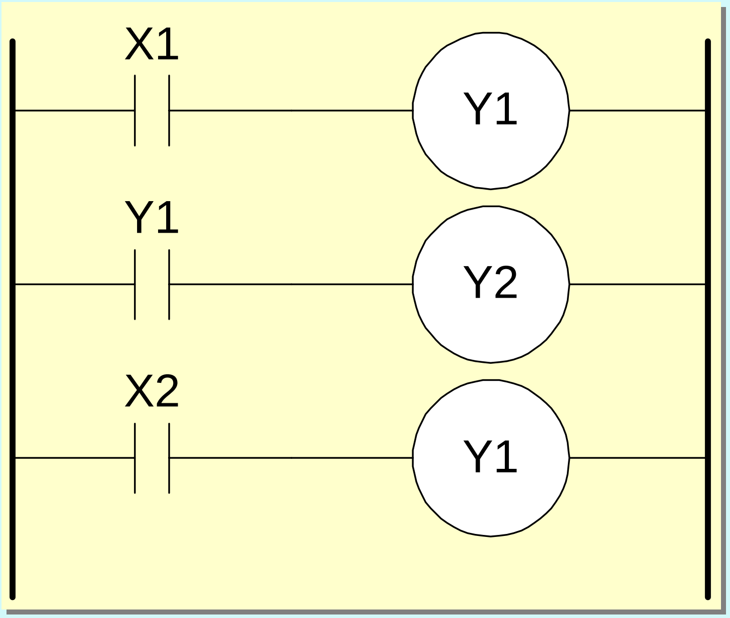
Input/Output updating:

In some PLCs, all <u>physical</u> inputs are updated at the beginning of the scan and all <u>physical</u> outputs updated at the end of the scan.

Program Scan – Double coiling

Double coiling, or specifying the same output twice, is allowed.

In figure, if X1=ON and X2=OFF, then Y2=ON and Y1=OFF.



Differences between Relay and PLC ladder diagrams

Relay circuit

Hardware components, relays and other switches, have a limited number of contacts.

For PLC

Software components, internal relays, inputs and outputs, have a "unlimited" number of contacts. No need to try to "save". Better to make program easier to read.

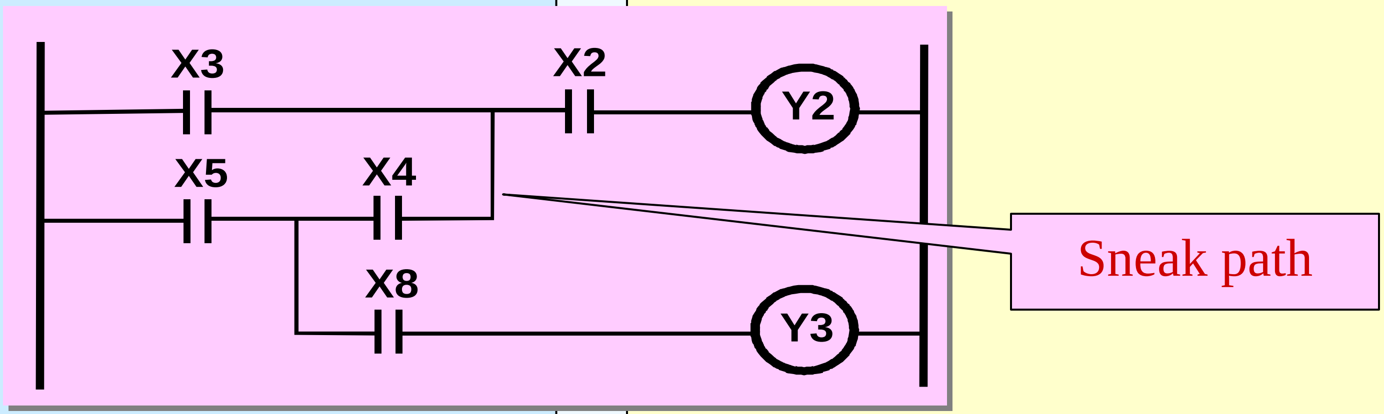
Differences between Relay and PLC ladder diagrams

Relay circuit

Current flow can take place in any direction .

For PLC

“Current flow" takes place only in one direction, from left to right.

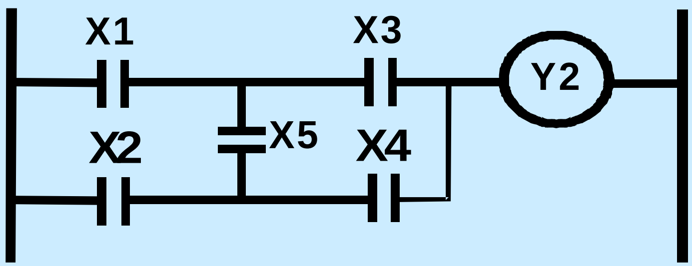


This ladder diagram will need to be modified for PLC implementation.

Differences between Relay and PLC ladder diagrams

Relay circuit

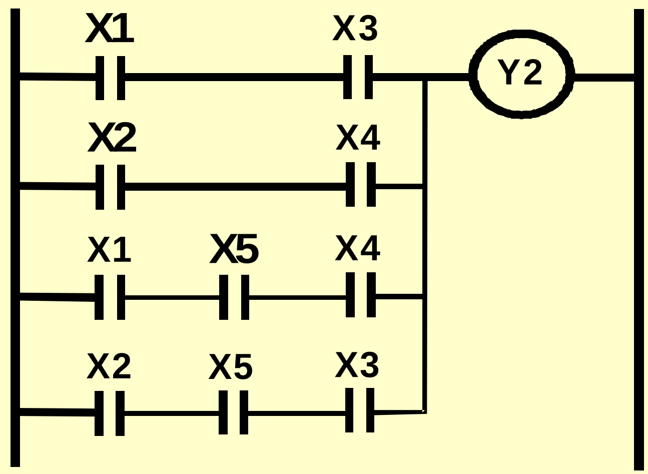
Contacts cannot be placed vertically , with crossover lines.



Relay circuit. Will need to be modified for PLC.

For PLC

Ladder diagrams are strictly two-dimensional and there can be no crossover lines.



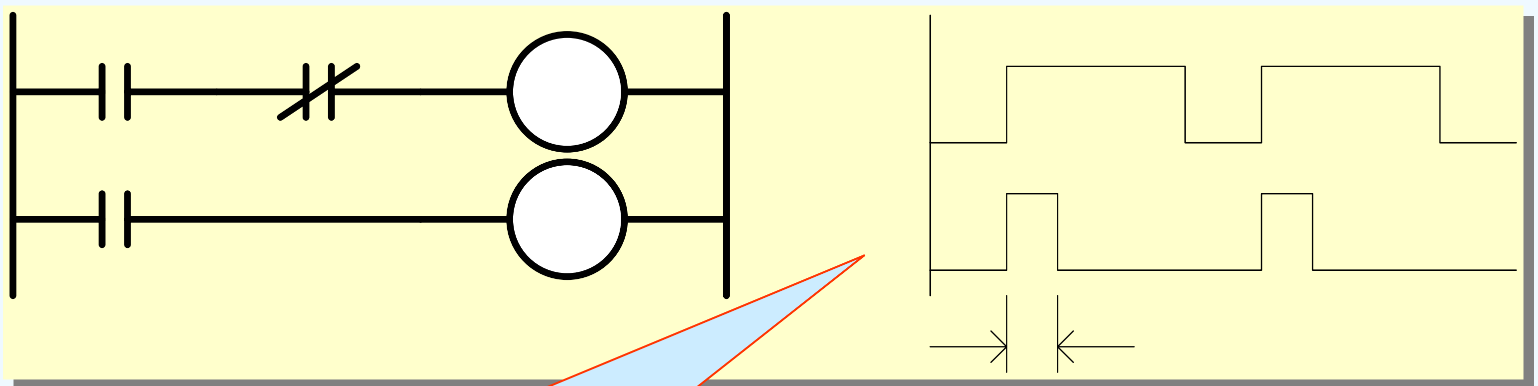
Equivalent PLC circuit

Differences between Relay and PLC ladder diagrams

Relay circuit	For PLC
All rungs of the ladder diagram are active simultaneously. This "parallel" operation sometimes causes "race" problems and malfunctions .	Each rung of the ladder is scanned, and acted upon, successively starting from the first rung. When the last rung has been scanned, a new cycle begins from the first rung again. Scanning period of the order of 5 to 50 ms
The order in which the rungs are drawn is immaterial.	The order in which the rungs are "programmed" into memory is very important.

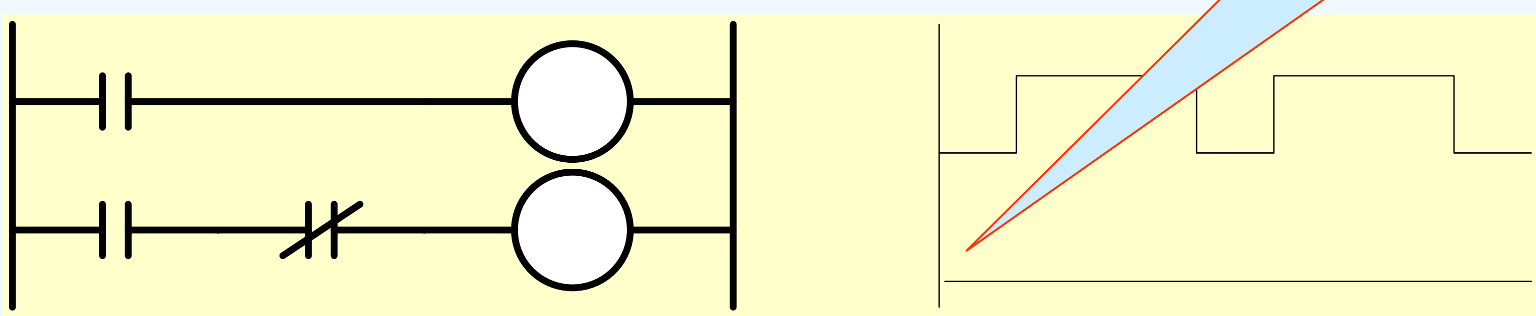
Program Scan – Program order

For the PLC, the order in which the rungs are "programmed" into memory is very important.



One output pulse for every positive transition of X3

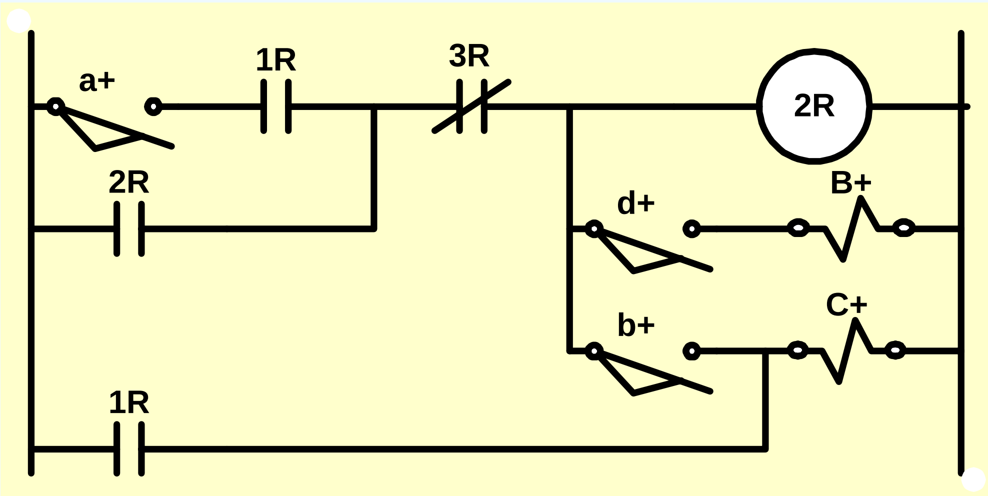
Y2 always OFF



X3

M1

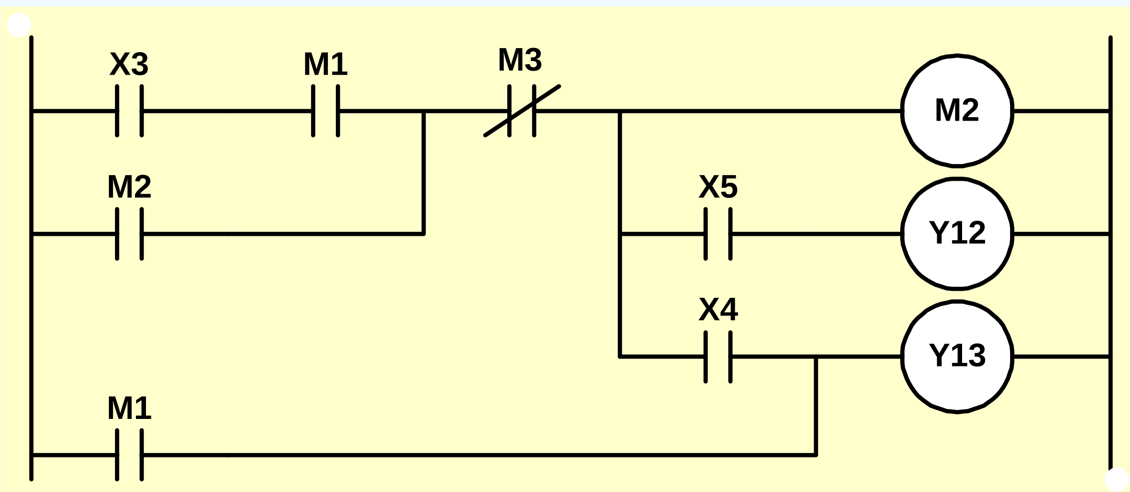
An Example



Original Ladder diagram

Input Module	Output Module	Connected to
X3		a+
X4		b+
X5		d+
	Y12	B+
	Y13	C+

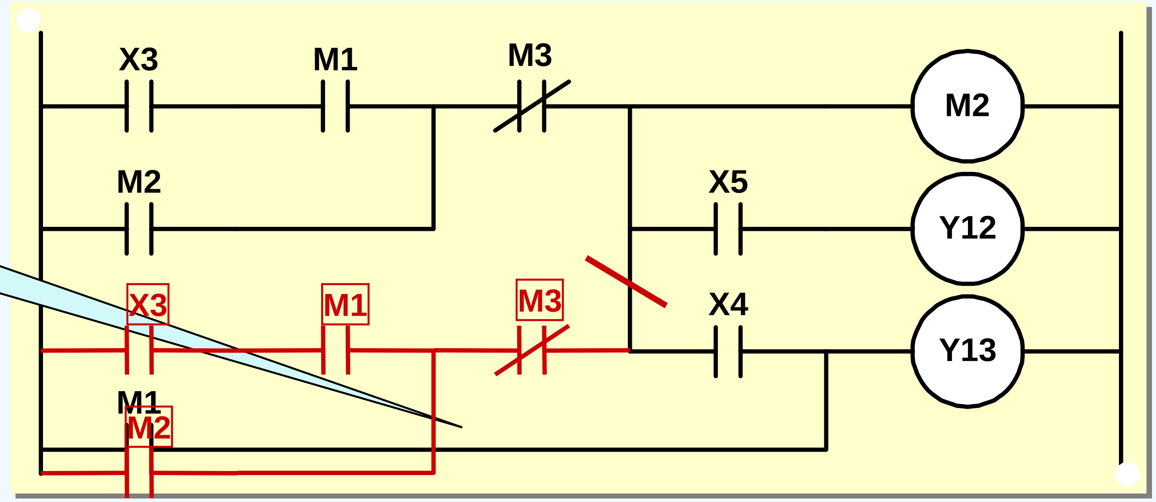
I/O allocation



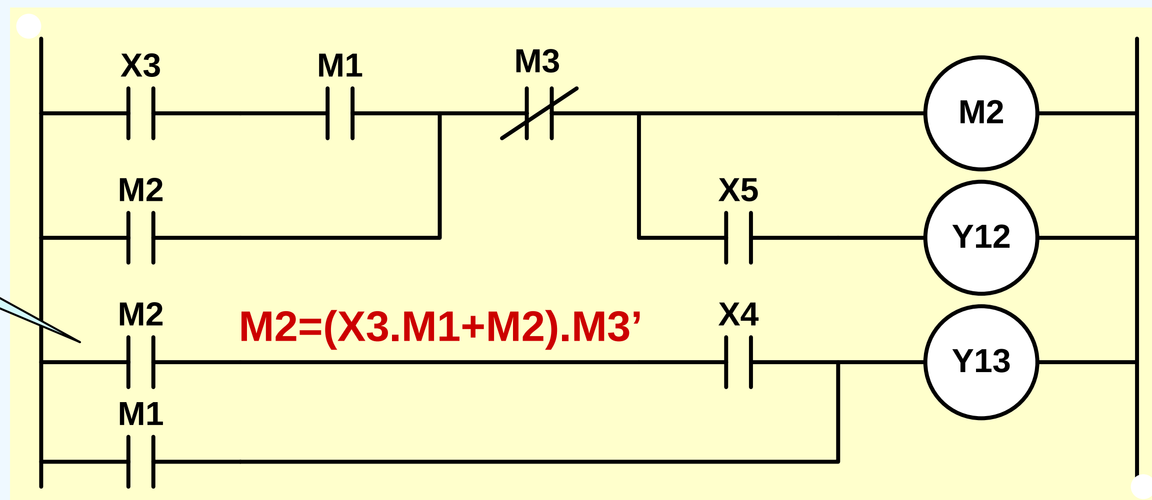
Re-drawn Ladder diagram
For PLC

An Example

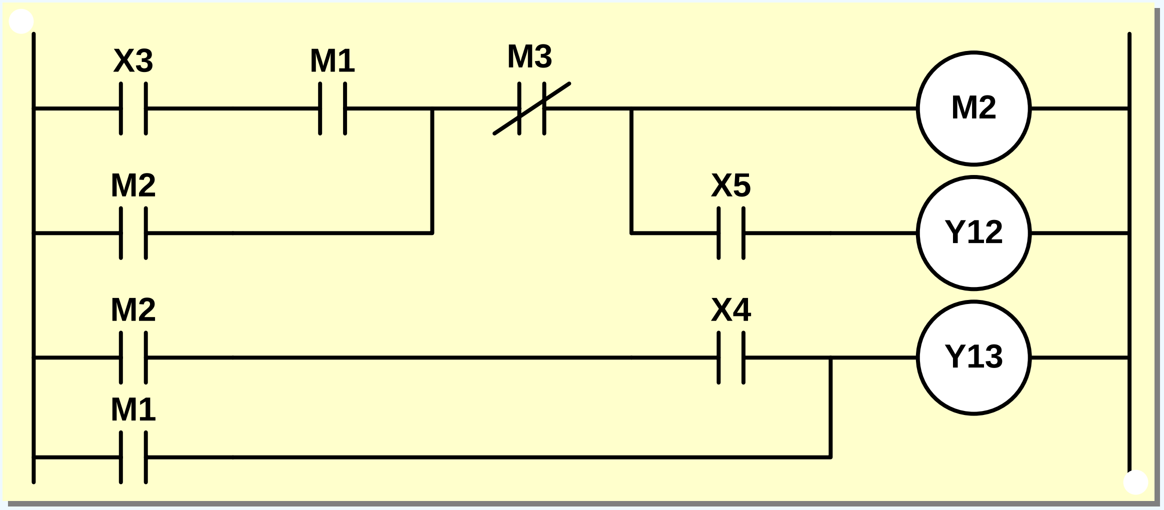
Cannot be programmed



Redrawn



An Example



Program

```
LD   X3
AND  M1
OR   M2
ANI  M3
OUT  M2
AND  X5
OUT  Y12
LD   M2
AND  X4
OR   M1
OUT  Y13
END
```

End of PLC