Relay + programmable Logic Controllers

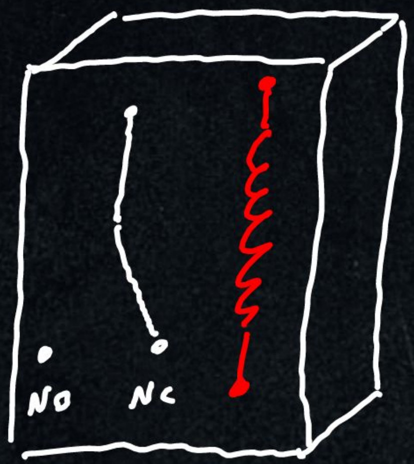
1) RLC
2) PLC

① RLC

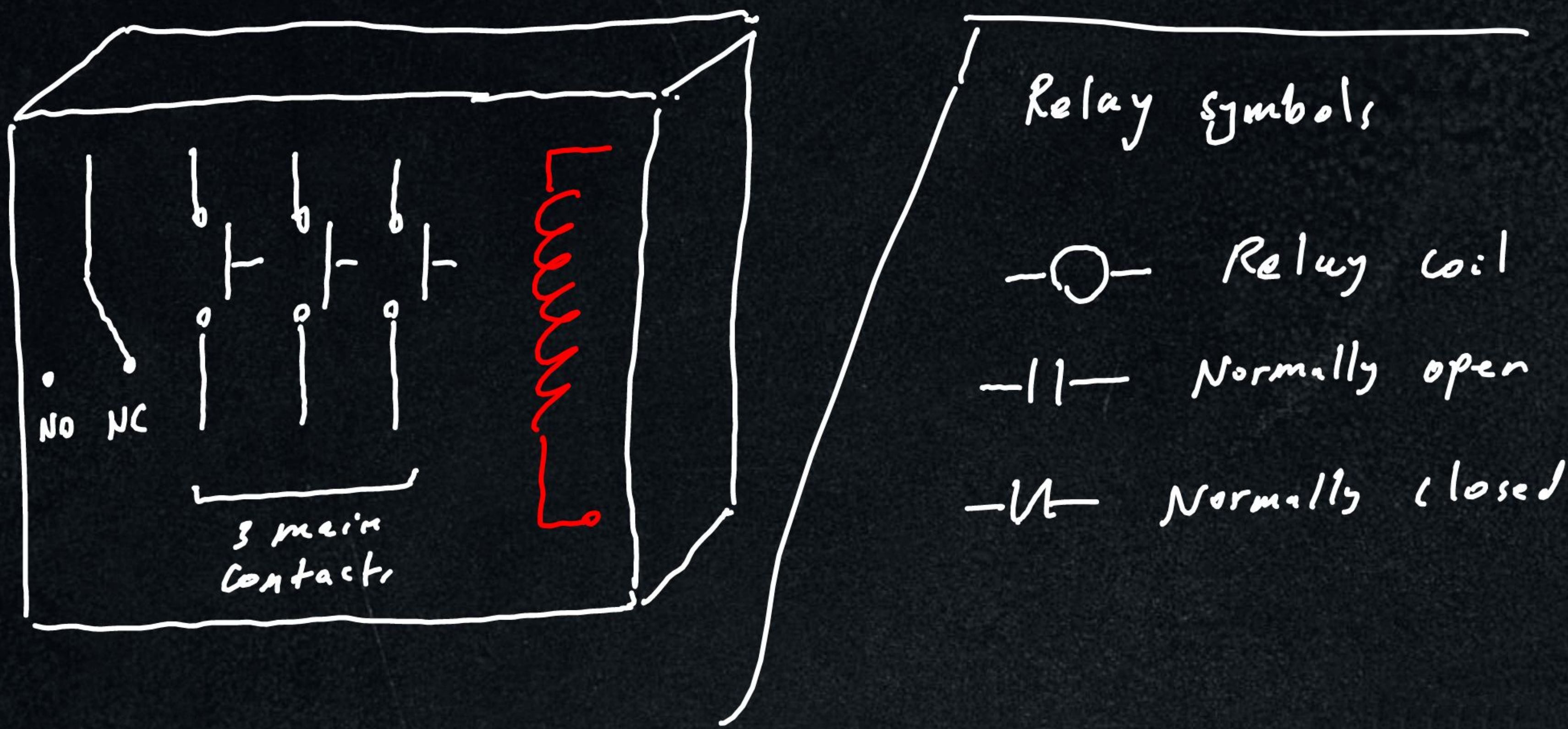
- prior to the widespread use of PLCs in process control and automation, hardwired relay control systems were used.
- Control of motors such as starting, stopping, reversing the direction, sequence starting, etc can be achieved by designing suitable control circuits using relays, contactors, push buttons, etc

Relay :- It is an electromagnet switch that has a coil and a set of associate contacts.

→ Contacts can be either normally open or normally closed. "Auxiliary contacts"



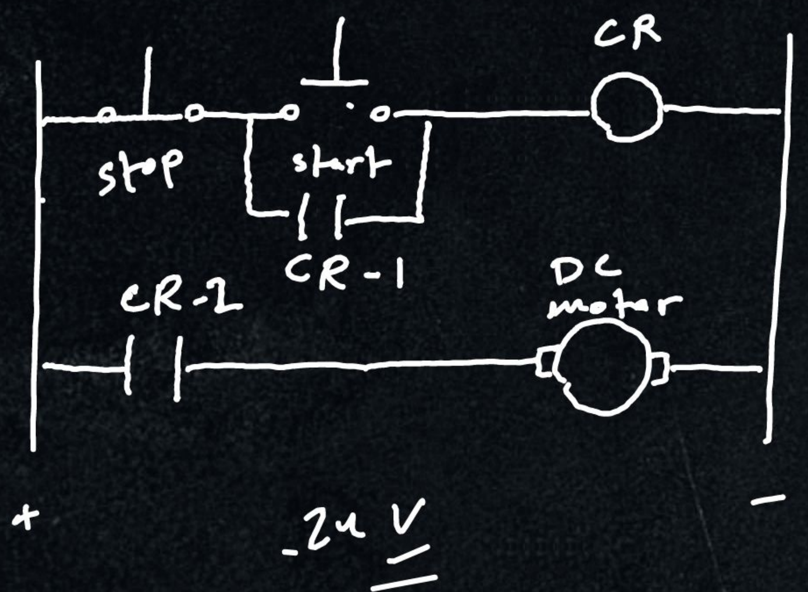
Contactor :- It is an electromagnet switch that has a coil and 3 main contacts with a few auxiliary contacts.



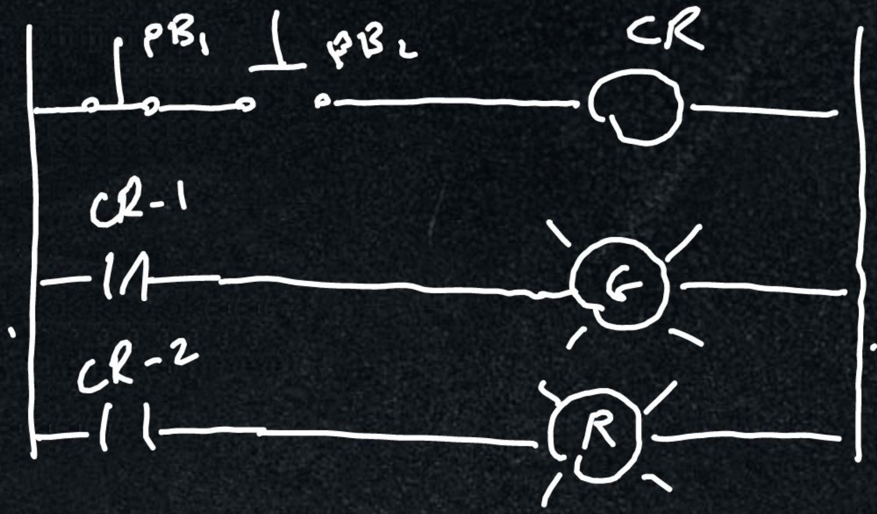
Line Diagram

Examples;

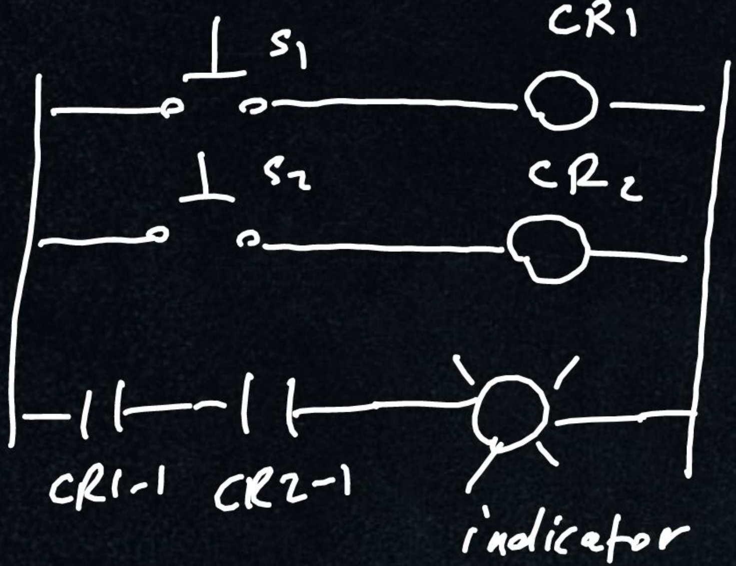
1) start / stop DC motor



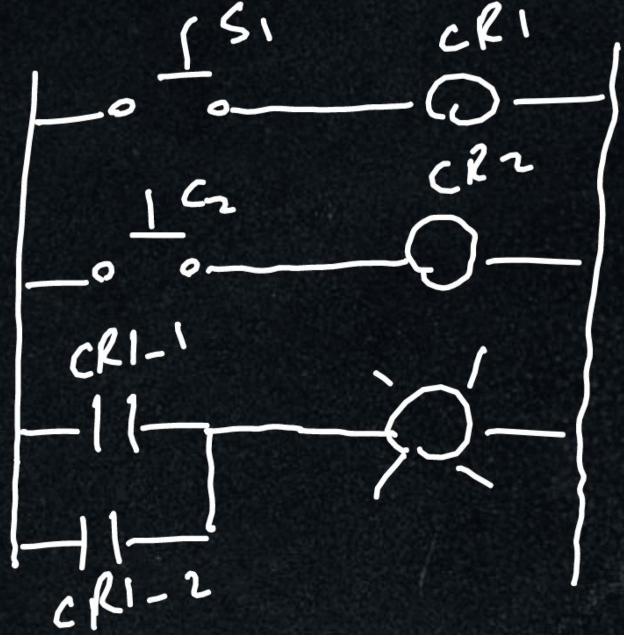
2) Relay controlling two pilot lights



3) AND logic $y = AB$

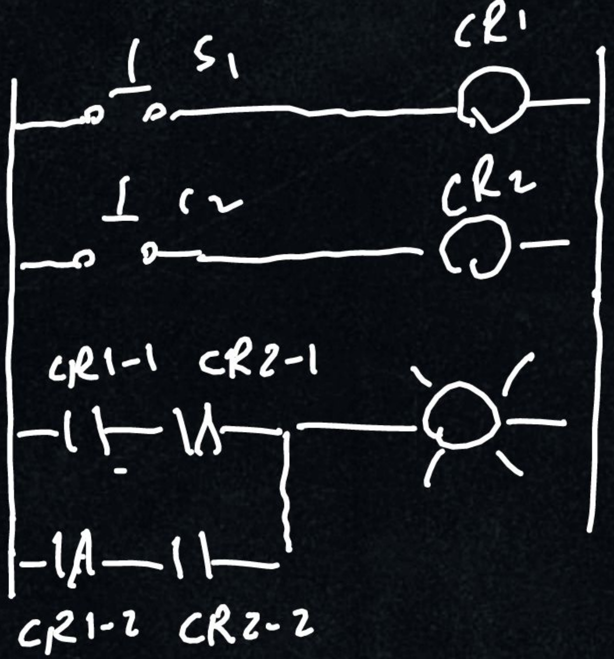


4) OR logic $y = A+B$



5) XOR $y = AB' + A'B$

5)

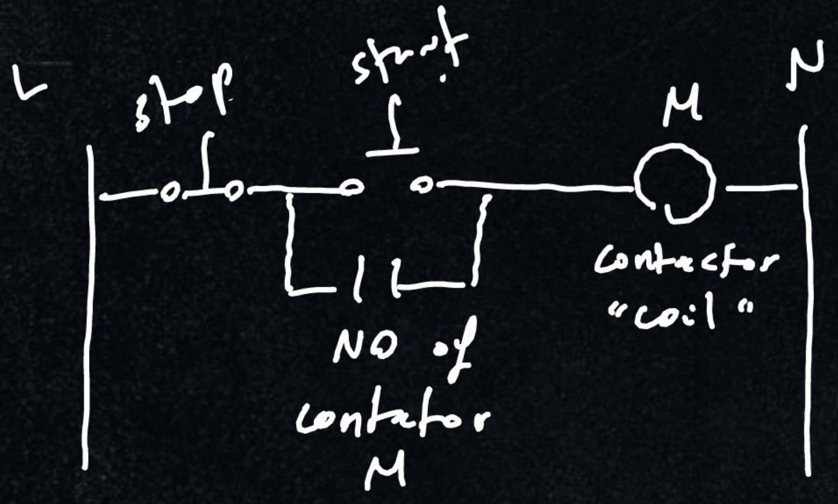


$$y = AB' + A'B$$

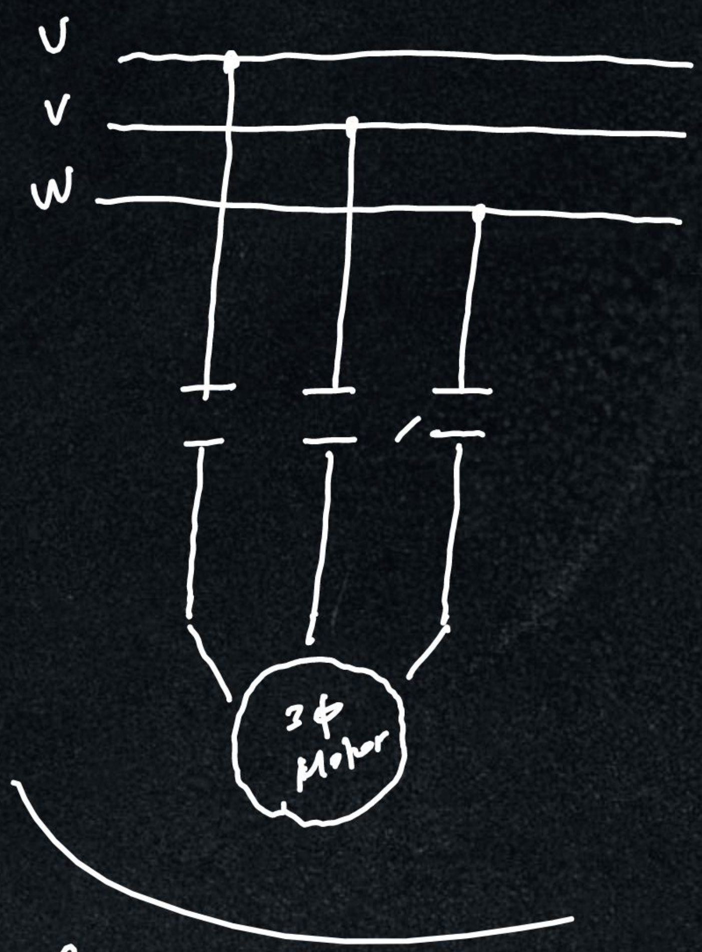
AND AND

OR

6) Start / stop of 3-ϕ Motor

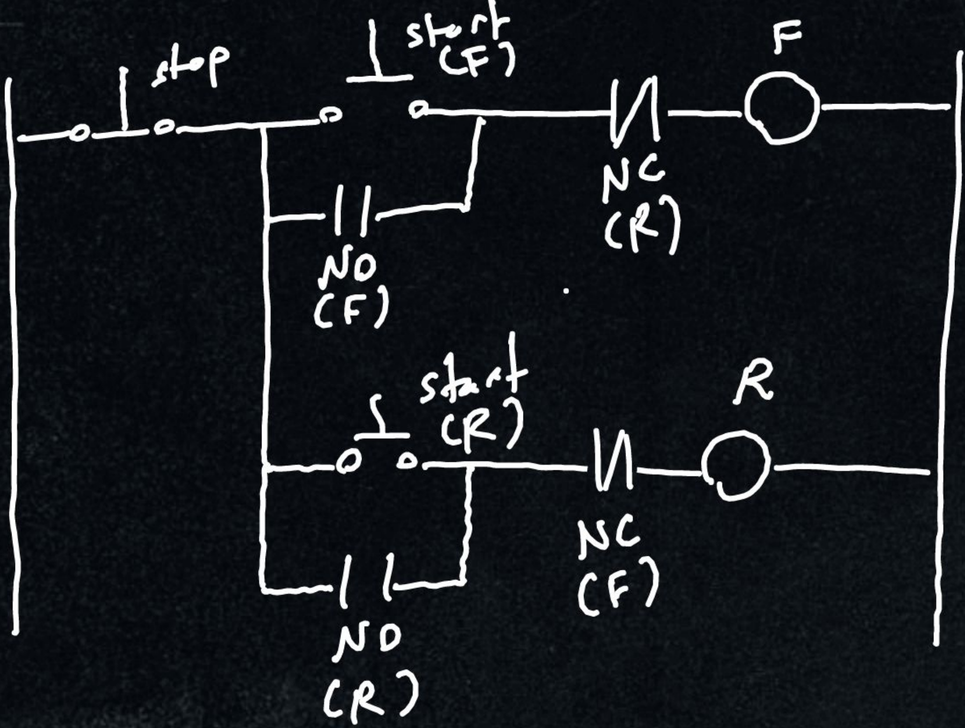


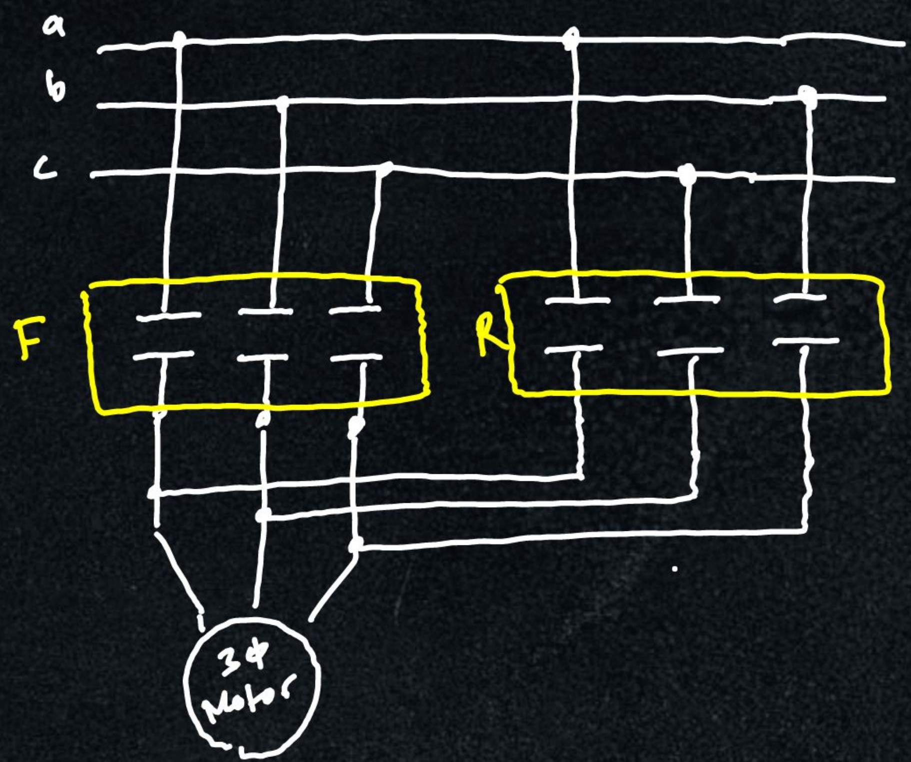
Control circuit



power circuit

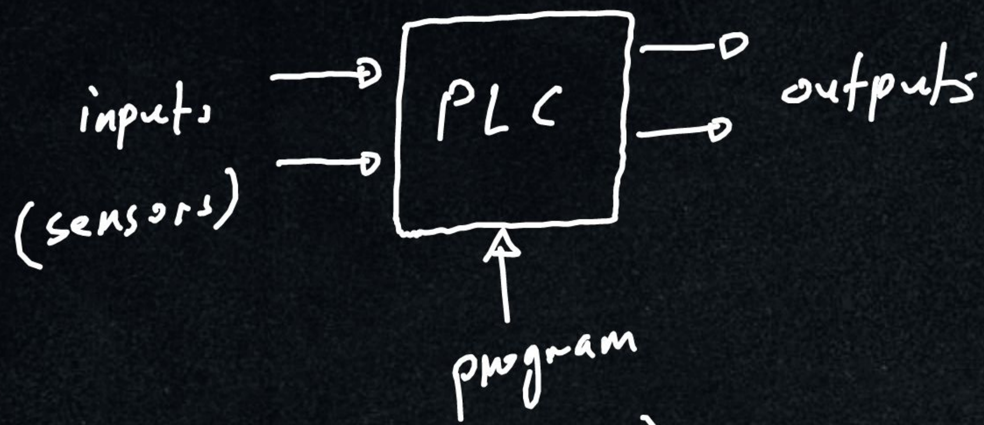
7) Reversing the direction of 3ϕ motor





② PLC

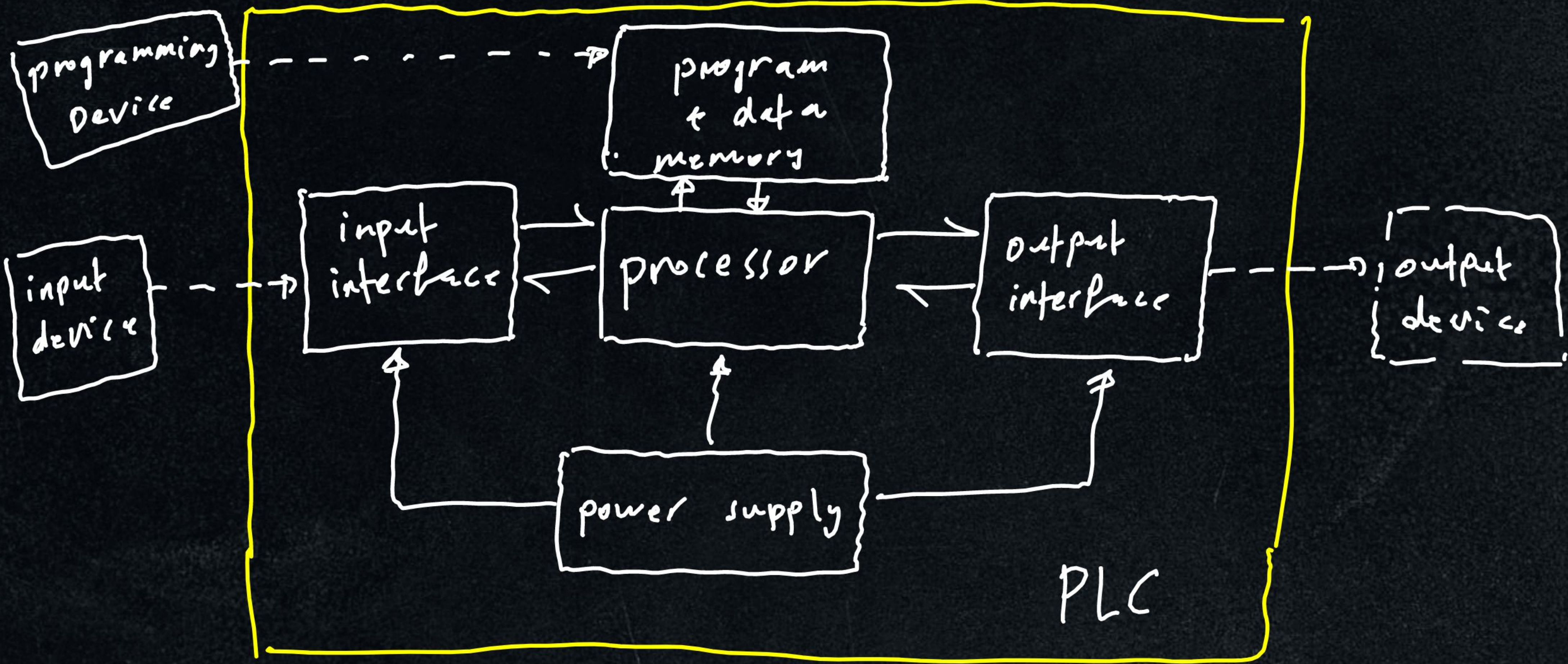
• It is a special form of microprocessor-based controller that uses memory to store instructions.



Advantages (compared to RLC)

1) can easily implemented changes "implemented in software"
2) More reliable
3) more compact
4) require less maintenance
5) can operate faster

Block Diagram



PLCs :-

* Mitsubishi
* Siemens
* Toshiba
* Allen - Bradley
* Schneider

Programming Languages

- Ladder diagrams (LAD)
- Instruction lists (IL)
- Function block diagram (FBD)
- structured text (ST)
- sequential function charts (SFC)

I/O Addresses

- The PLC has to be able to identify each input and output. It does this by allocating addresses to each input + output.
- This is likely to be just a number prefixed by a letter to indicate whether it is an input or output.
- Mitsubishi

24 inputs X400 - X427 in octal format

X400, X401, X402, X403, X404, X405, X406, X07, X410, ...

16 outputs Y430 - Y447

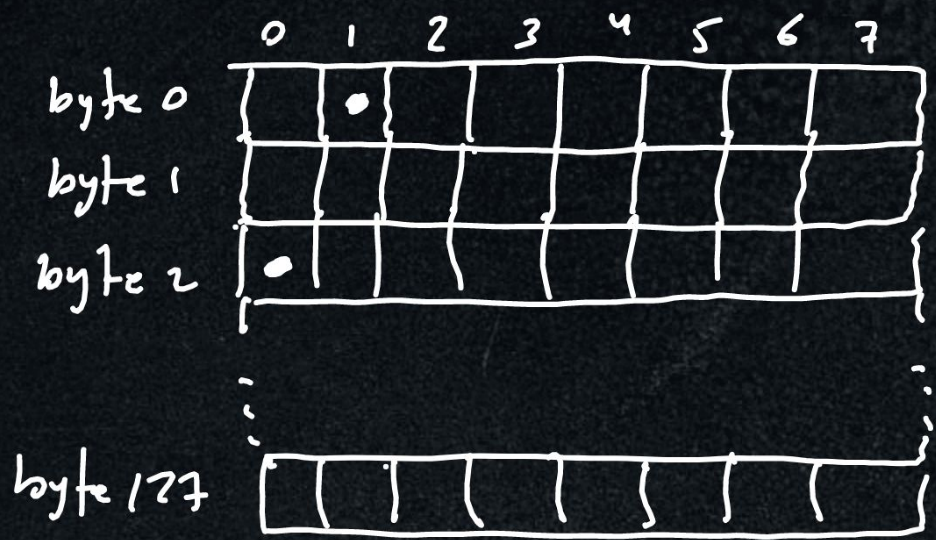
• Siemens

The digital I/O is arranged into groups of 8 bits, called a byte. A signal is identified by its bit number (0-7) and its byte number (0-127)

X XX.X → bit number

X → I : input, O : output

XX → byte number



EX :-

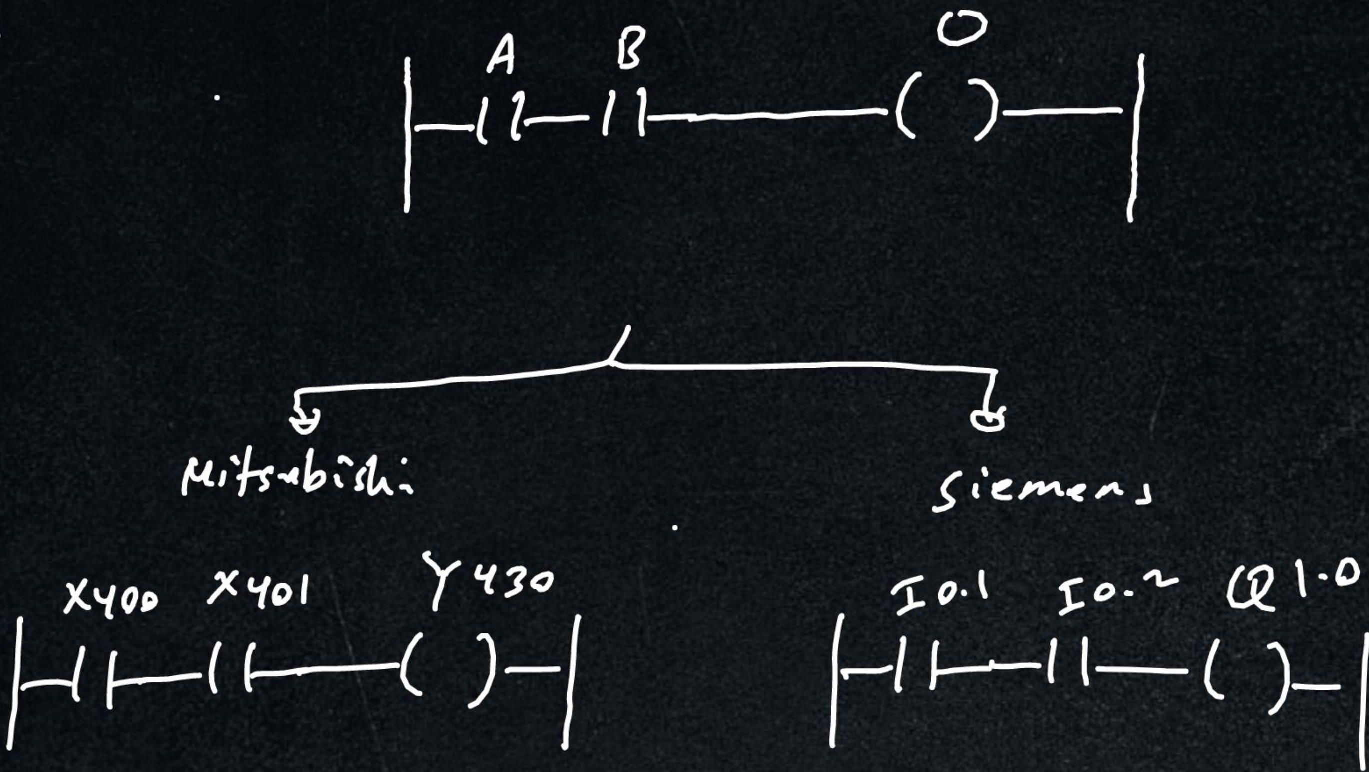
I 0.1 input at bit 1 in byte 0

O 2.0 output at bit 0 in byte 2

Logic Functions (LAD)

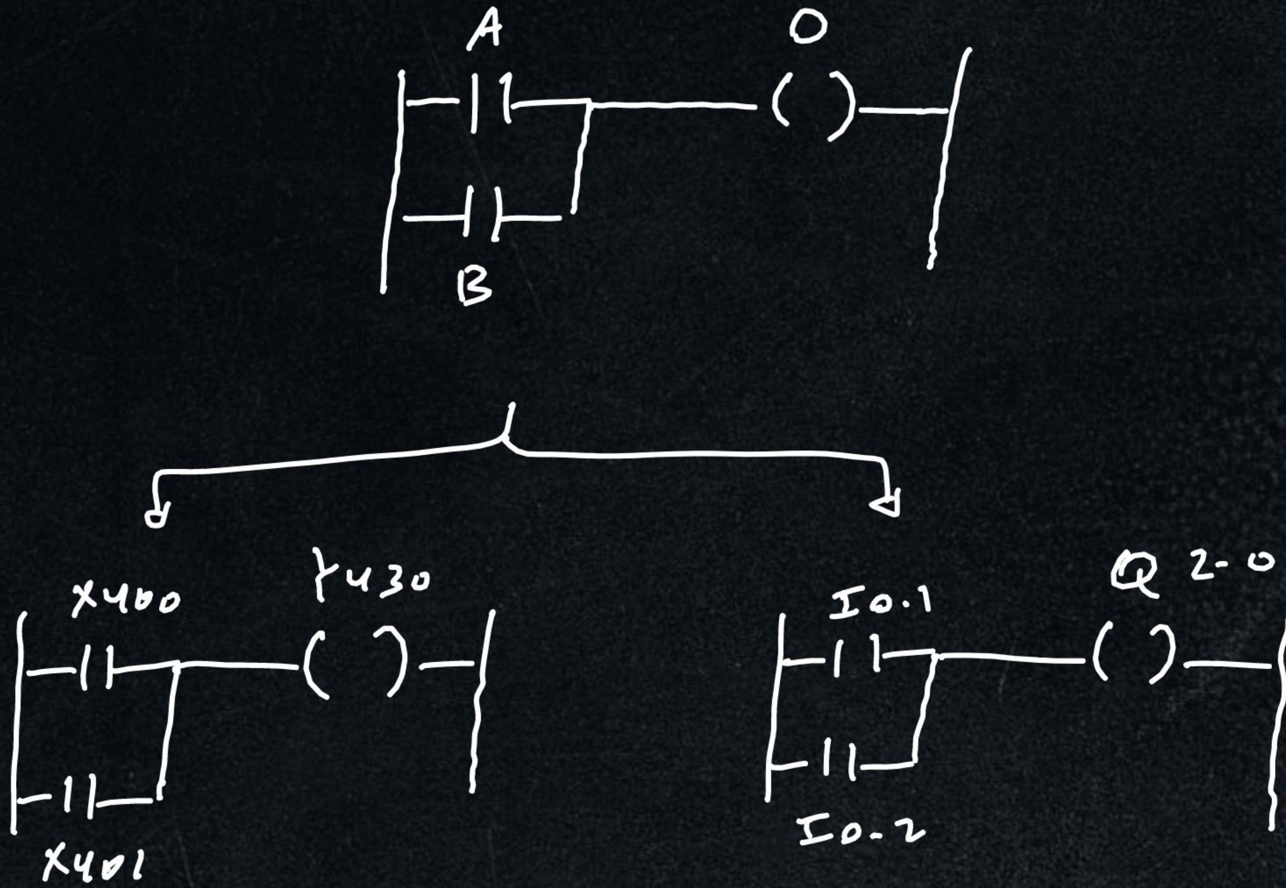
1) AND $O = AB$

A	B	O
0	0	0
0	1	0
1	0	0
1	1	1



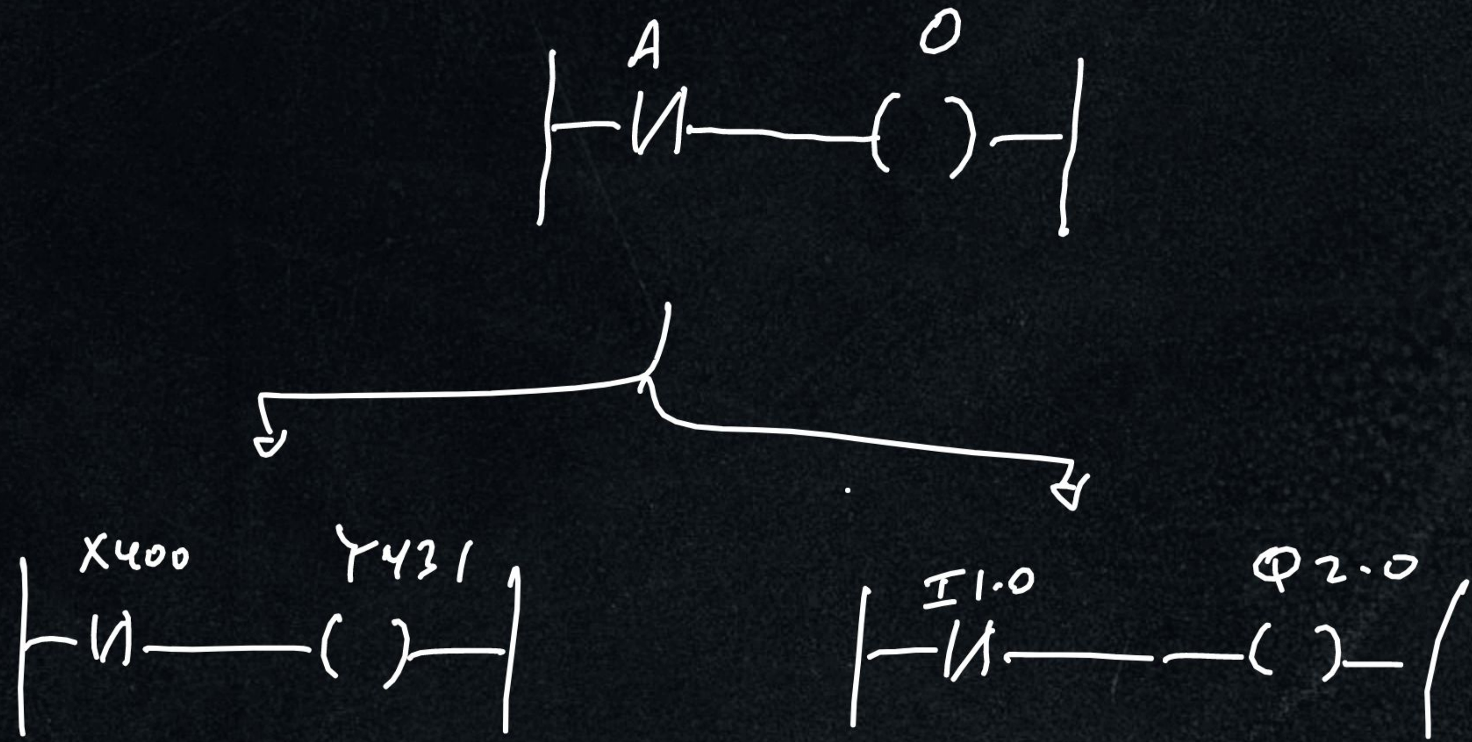
2) OR $O = A + B$

A	B	O
0	0	0
0	1	1
1	0	1
1	1	1



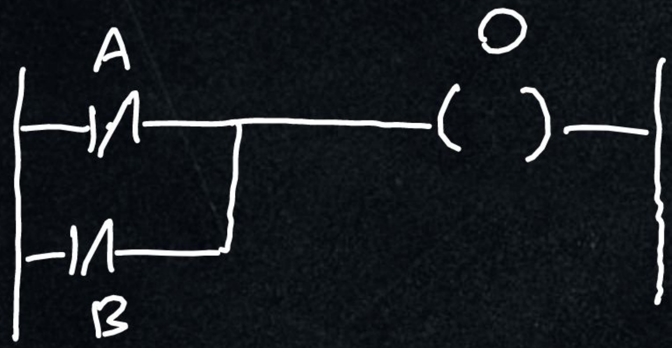
3) NOT $O = A'$

A	O
0	1
1	0



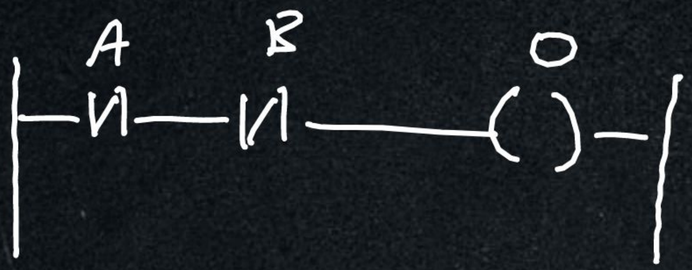
4) NAND $O = (AB)' = A' + B'$

A	B	O
0	0	1
0	1	1
1	0	1
1	1	0



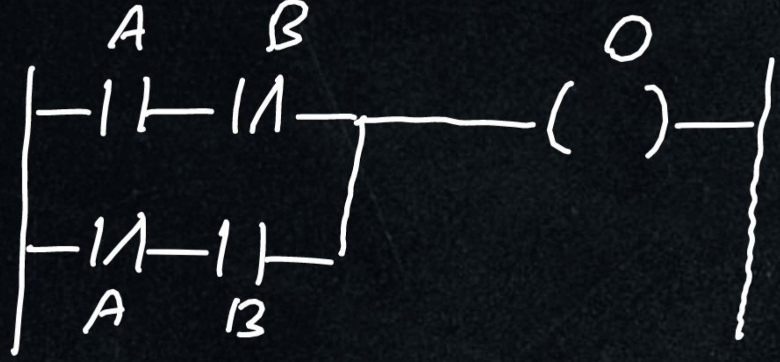
5) NOR $O = (A+B)' = A'B'$

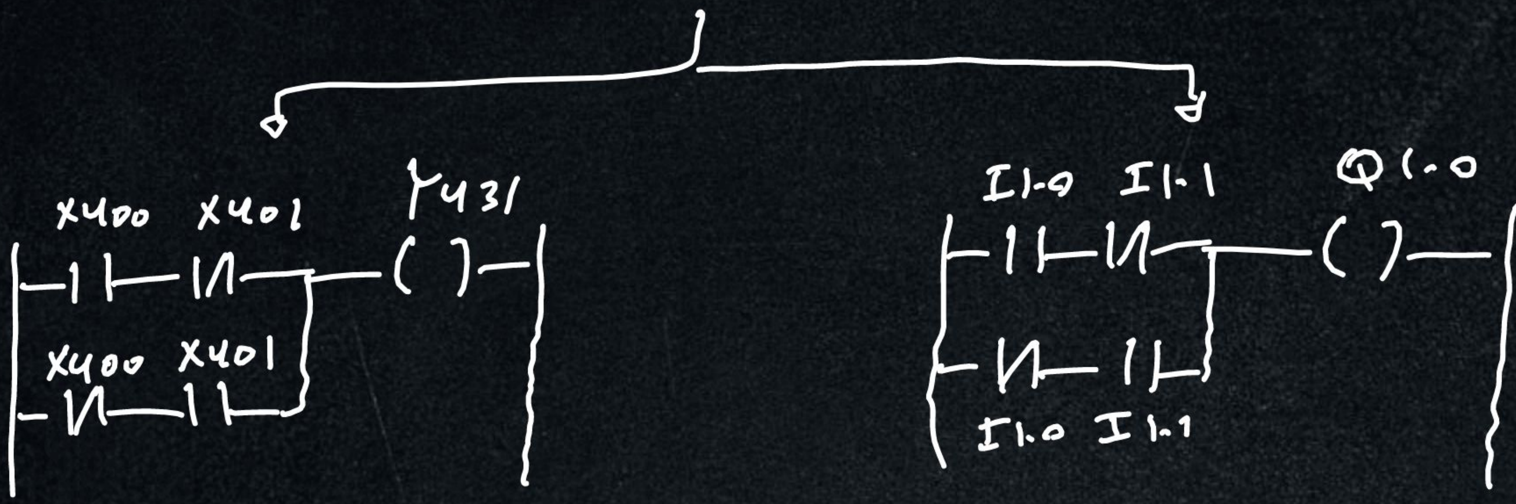
A	B	O
0	0	1
0	1	0
1	0	0
1	1	0



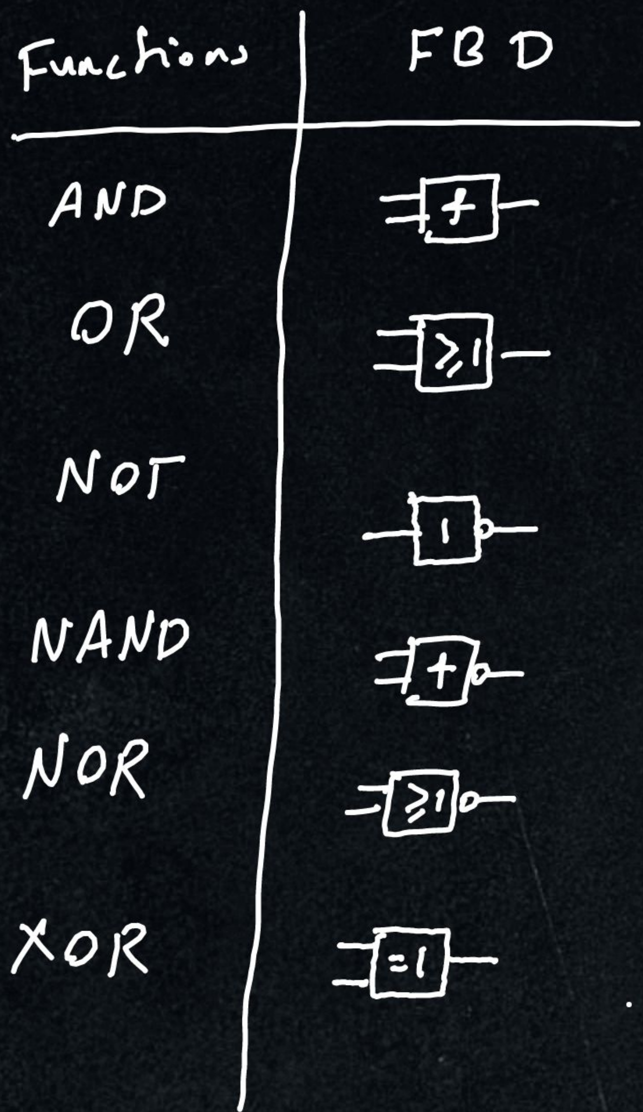
6) XOR $O = AB' + A'B$

A	B	O
0	0	0
0	1	1
1	0	1
1	1	0

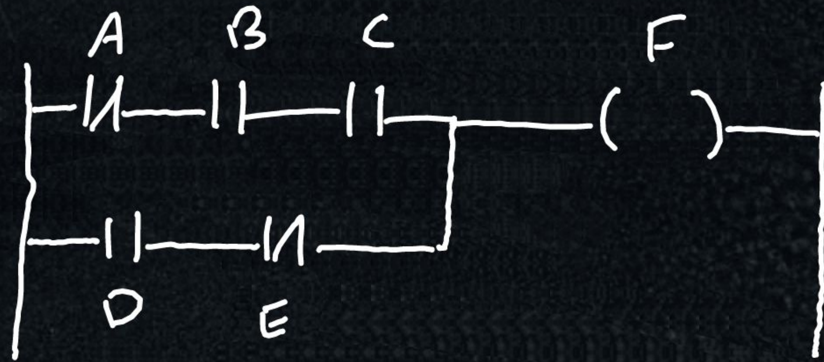




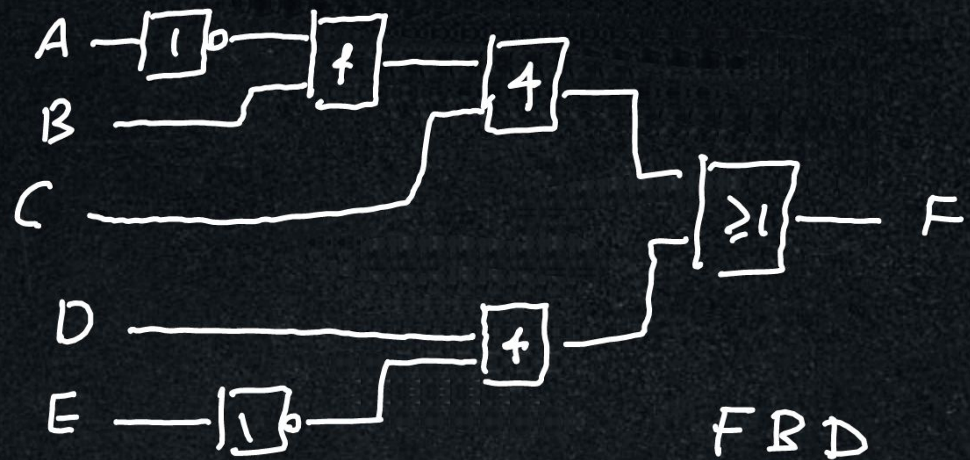
Logic Functions (FBD)



EX :



LAD



FBD

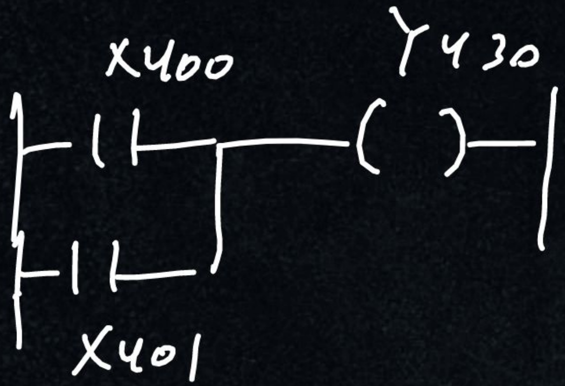
$$F = A'BC + DE'$$

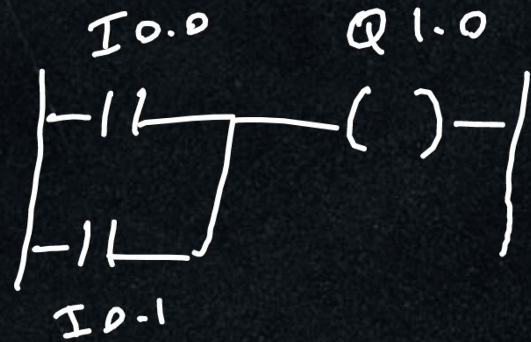
IL programming

Mitsubishi	Siemens	
LD	A	start a rung with open contacts
LDI	AN	" " " " closed "
AND	A	series element with open contacts
ANI	AN	" " " closed "
OR	O	parallel " " open "
ORI	ON	" " " closed "
OUT	=	An output.

Logic Functions (IL)

1) OR



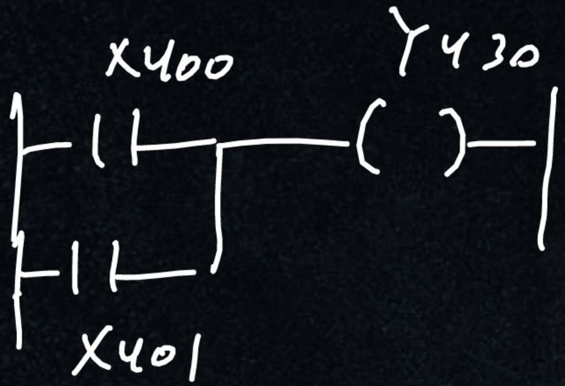


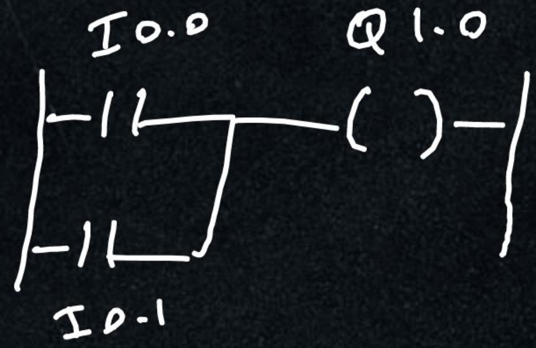

```
LD X400
OR X401
OUT Y430
```

```
A I0.0
O I0.1
= Q1.0
```

Logic Functions (IL)

1) OR





LD X 400

OR X 401

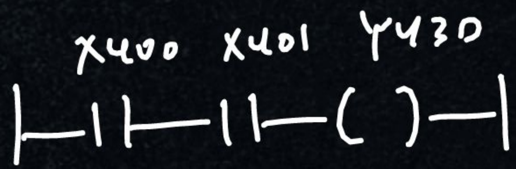
OUT Y 430

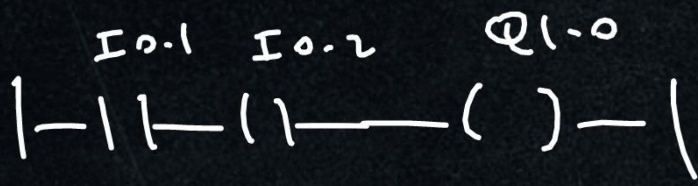
A I 0.0

O I 0.1

= Q 1.0

2) AND





LD X400

AND X401

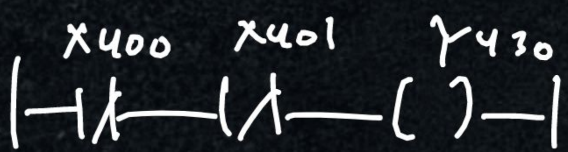
OUT Y430

A I0.1

A I0.2

= Q1.0

3) NOR $O = (A+B)' = A'B'$



LDI X400

ANI X401

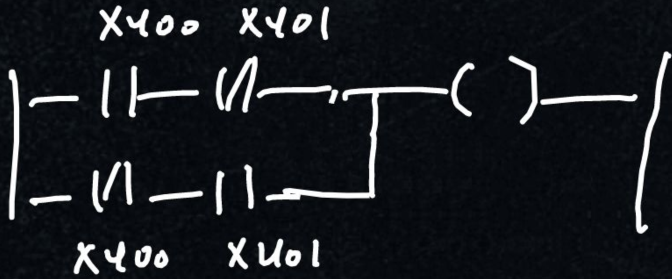
OUT Y430

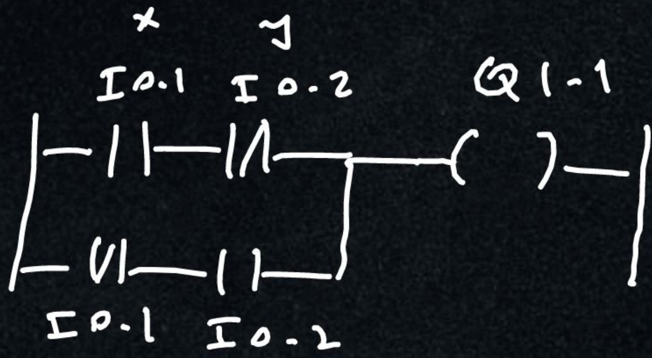
AN I0.1

AN I0.2

= Q2.0

4) XOR $O = AB' + A'B$





LD X400

ANI X401

LDI X400

AND X401

ORB

OUT Y430

A (

A I 0.1

AN I 0.2

)

O (

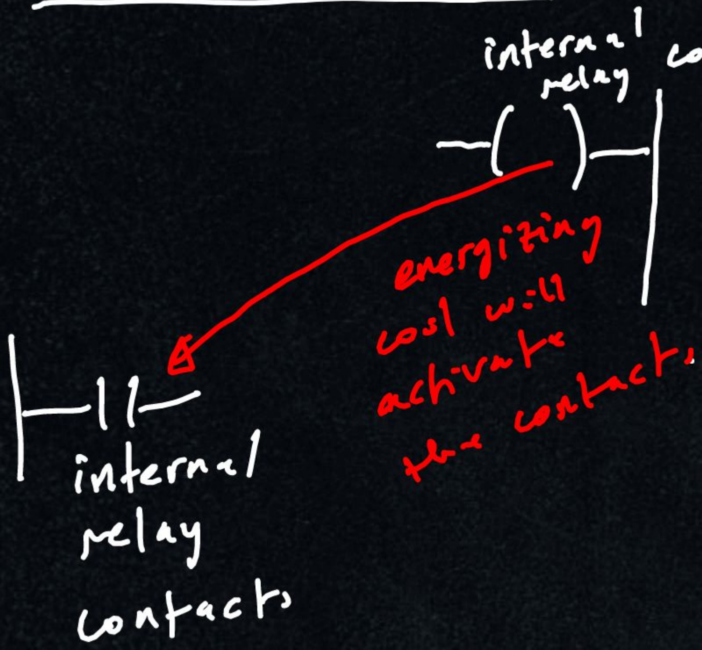
AN I0.1

A I0.2

)

= Q 1.1

Internal Relay

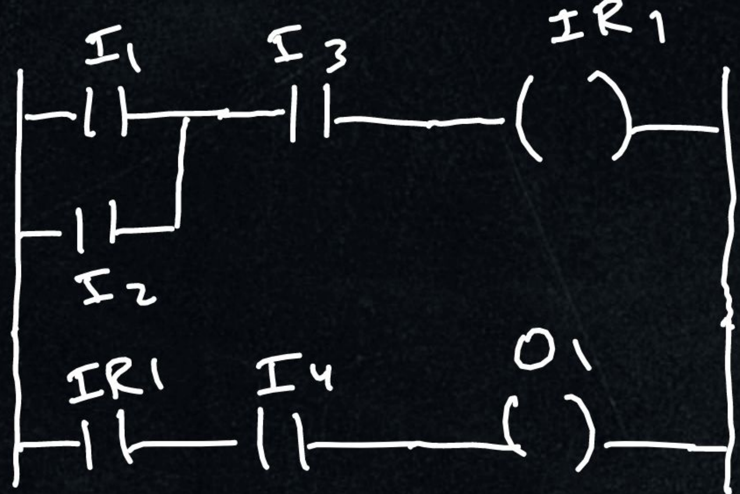


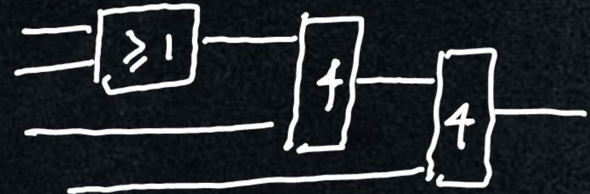
Addresses :-

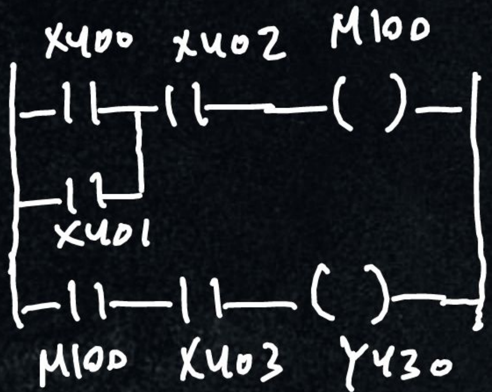
Mitsubishi M100, M101, M102, ...

Siemens F0.0, F0.1, F0.2, ...

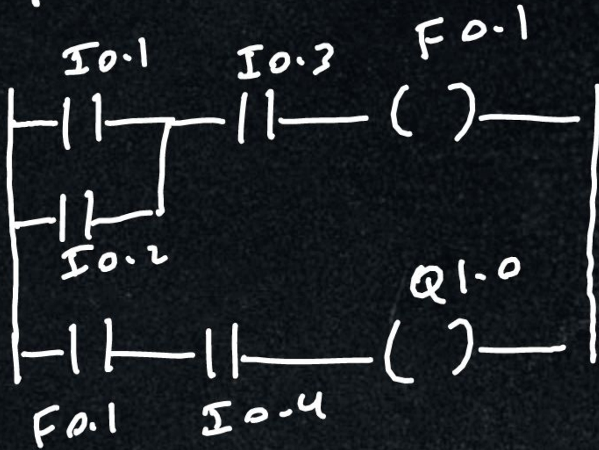
EX





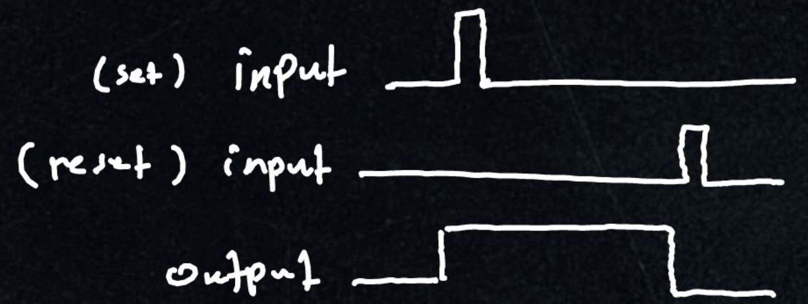



```
LD X400
OR X401
AND X402
OUT M100
LD M100
AND X403
OUT Y430
```

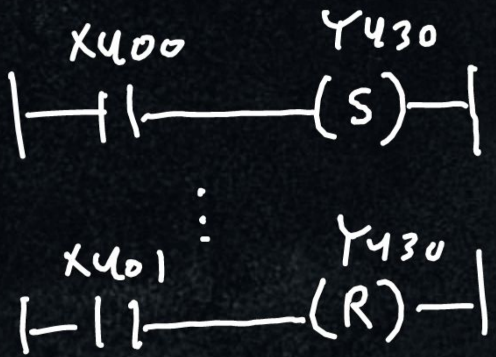



```
A I0.1
O I0.2
A I0.3
= F0.1
A F0.1
A I0.4
= Q1.0
```

Set and Reset

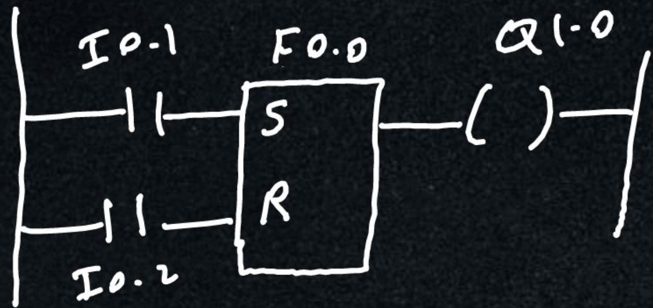


Mitsubishi




```
LD X400
S  Y430
 ⋮
LD X401
R  Y430
```

Siemens



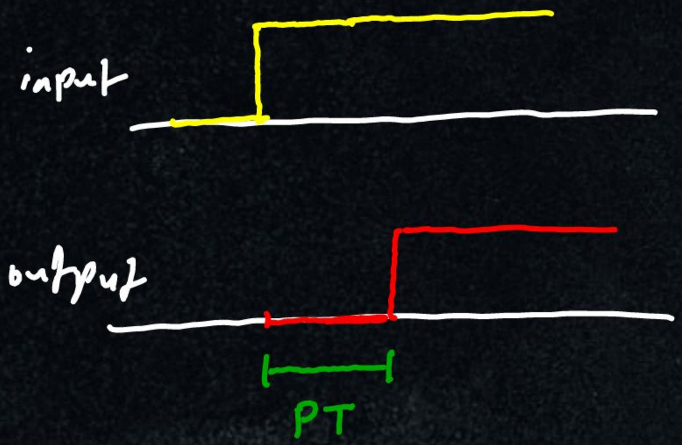

```
A I0.1
S F0.0
A I0.2
R F0.0

A F0.0
= Q1.0
```

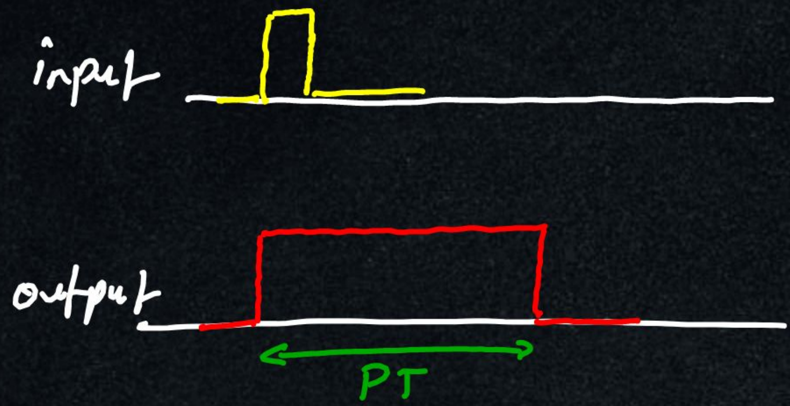
Timers

Types of timers

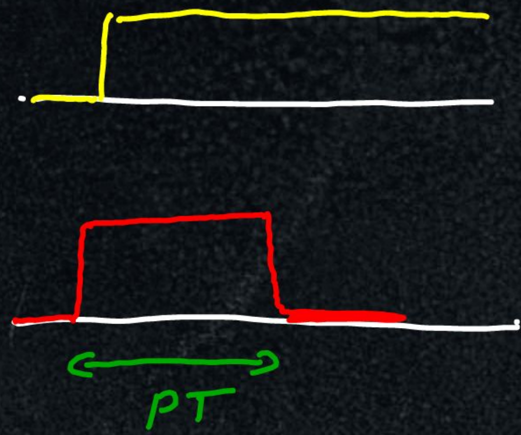
1) ON-delay timer



2) OFF-delay timer



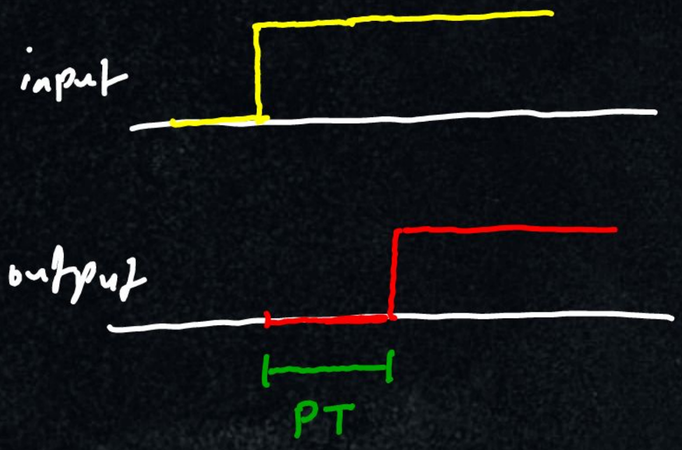
3) pulse timer



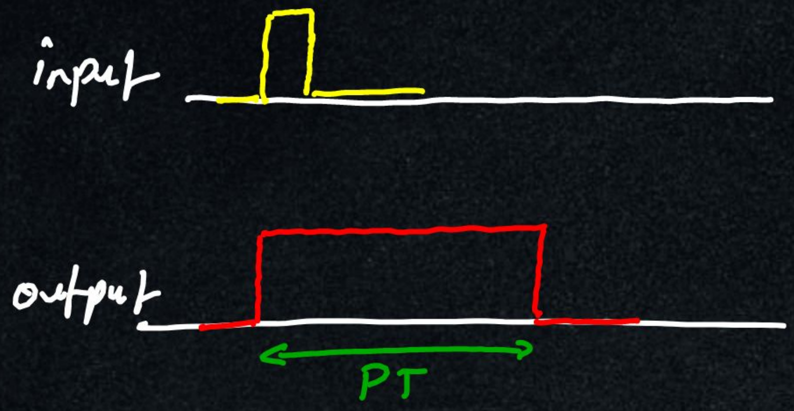
Timers

Types of timers

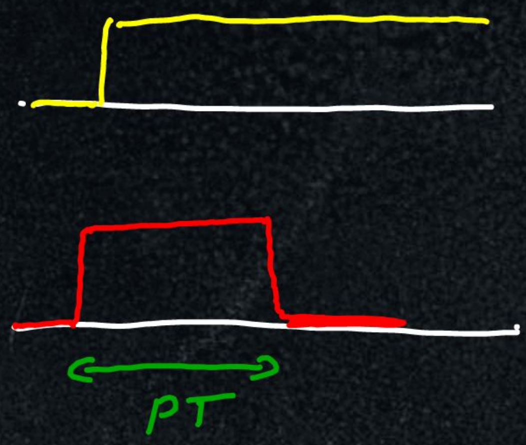
1) ON-delay timer



2) OFF-delay timer

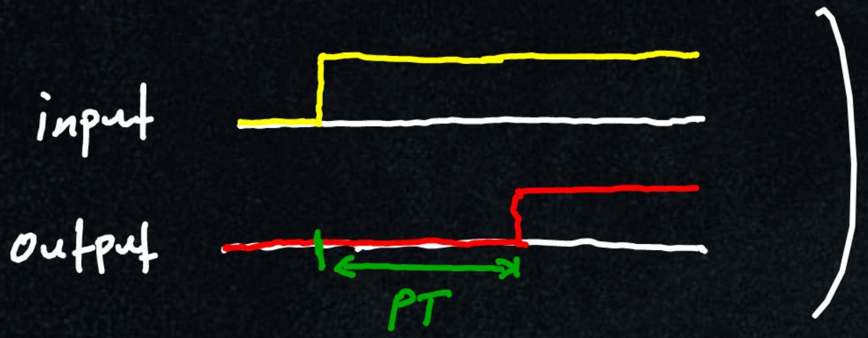


3) pulse timer



Timers

ON delay timer
OFF delay timer
pulse timer



ON-delay timer

PT: preset time

Mitsubishi

PT = n * time base of PLC

n is real integer

Time base:
- 10 ms
- 100 ms
- .1 sec
- 10 sec
- 100 sec

K5 ⇒ PT = 5 sec time base = 1 sec

K4 ⇒ PT = 4 sec time base = 1 sec

Siemens

KT xxx . x

xxx	x	time base
000	0	→ 10 ms
001	1	→ 100 ms
⋮	2	→ 1 sec
999	⋮	

KT 5.1

PT = 5 * 100 msec = 500 ms

KT = 7.2

PT = 7 x 1 sec = 7 sec

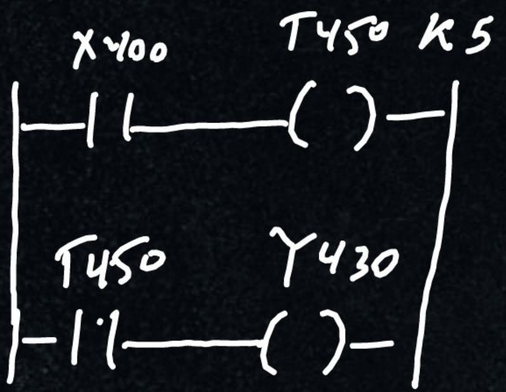
Addresses

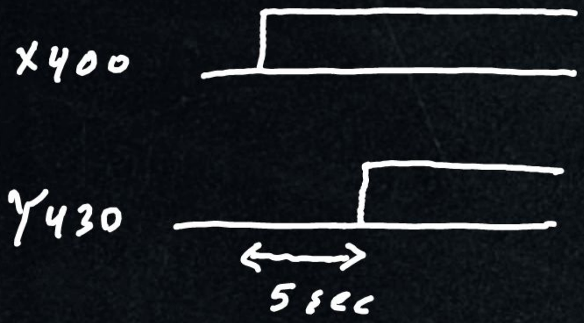
Mitsubishi : T450, T451, ...

Siemens : T0, T1, T2, ...

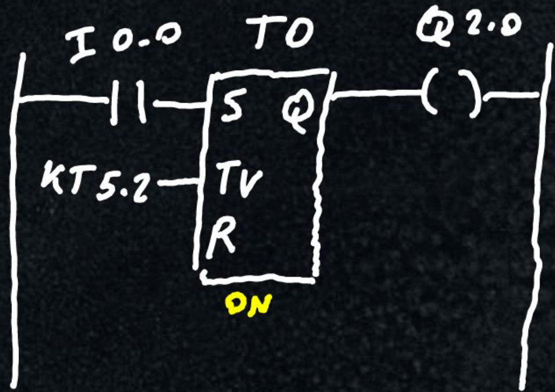
① ON-delay timer

Mitsubishi

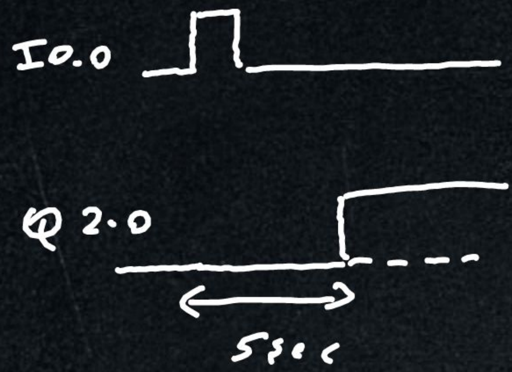




Siemens



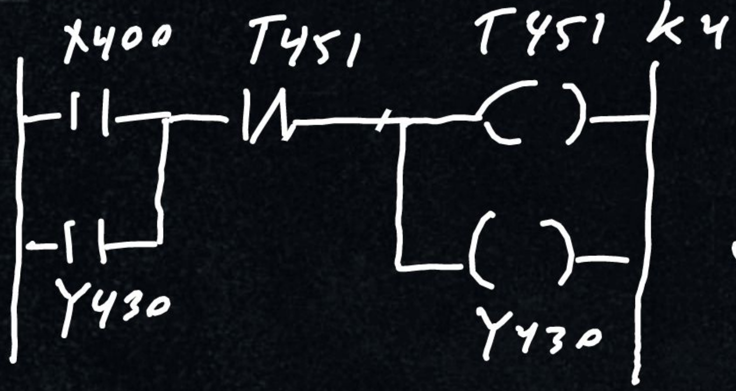
LD X 400
OUT T 450
K5
LD T 450
OUT Y 430

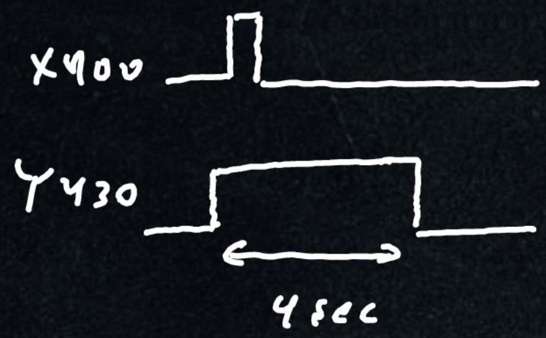


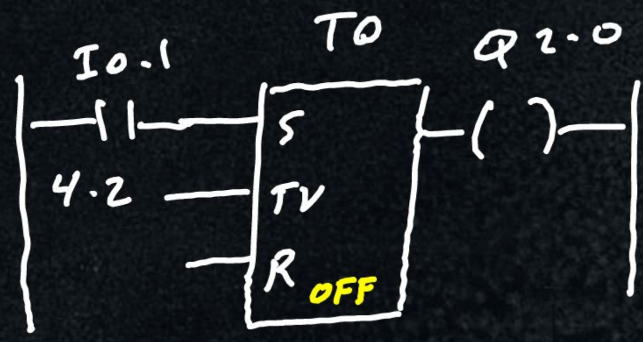
A I 0.0
SR T0
LKT 5.2
A T0
= Q 2.0

② OFF delay timer

timer → coil ←
timer → contact.

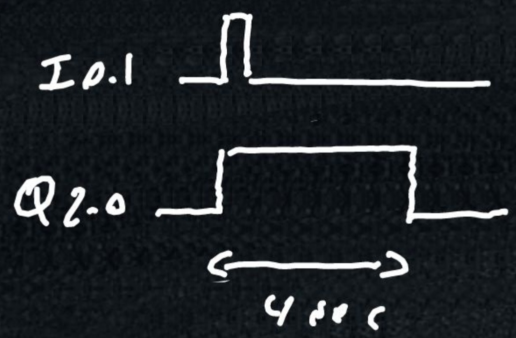




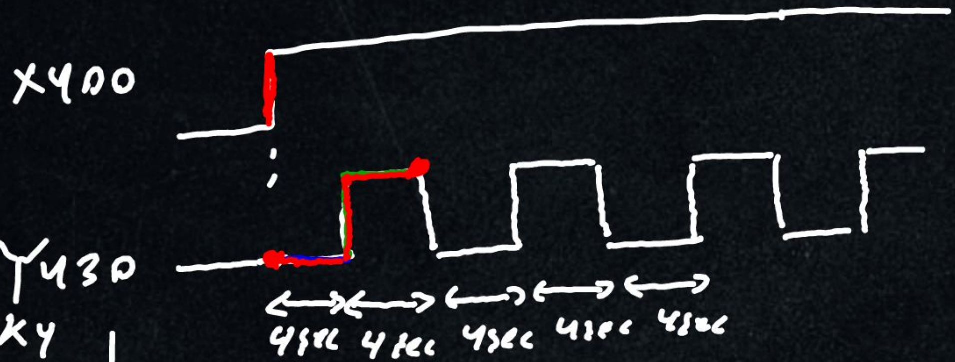


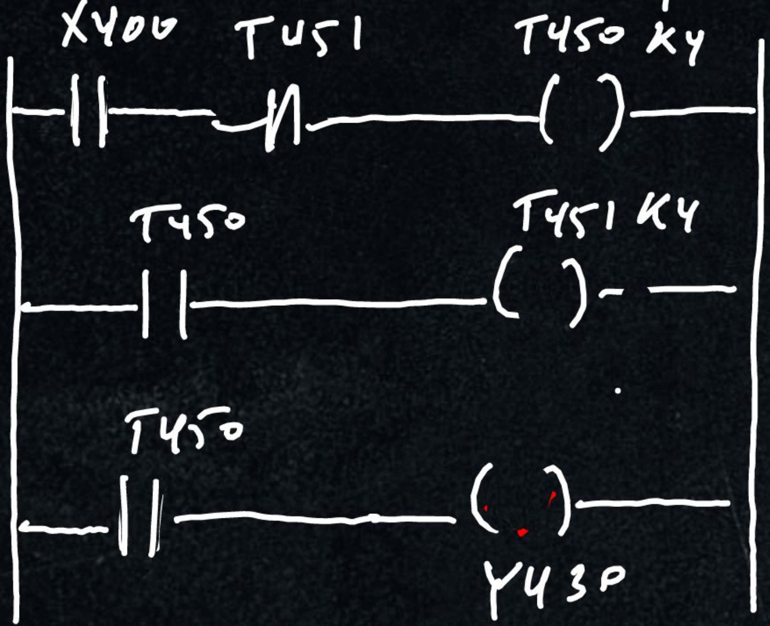

```
LD X400
OR Y430
ANI T451
OUT T451
  K4
LD X400
OR Y430
ANI T451
OUT Y430
```

```
A I 0.1
SR TO
LKT 4.2
A TO
= Q2.0
```



EX :- ON/OFF cycle





LD X400

ANI T451

OUT T450

K4

LD T450

OUT T451

K4

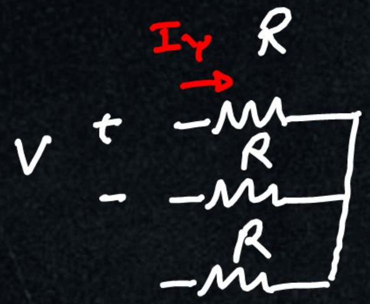
LD T450

OUT Y430

EX:- Y→Δ starter of induction motor

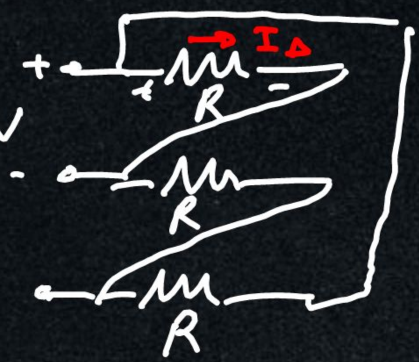
$$I_Y = \frac{\text{phase voltage}}{R}$$

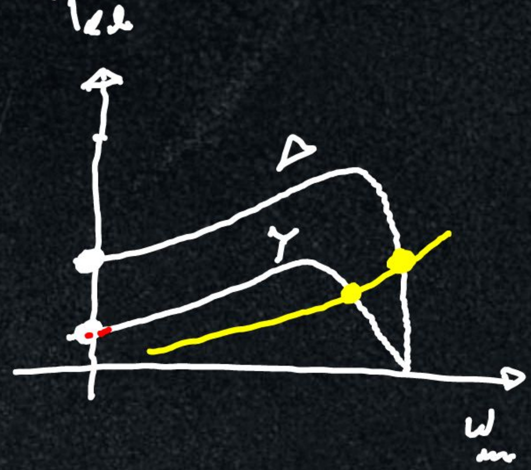
$$I_Y = \frac{V/\sqrt{3}}{R}$$



$$I_\Delta = \frac{\text{phase voltage}}{R}$$

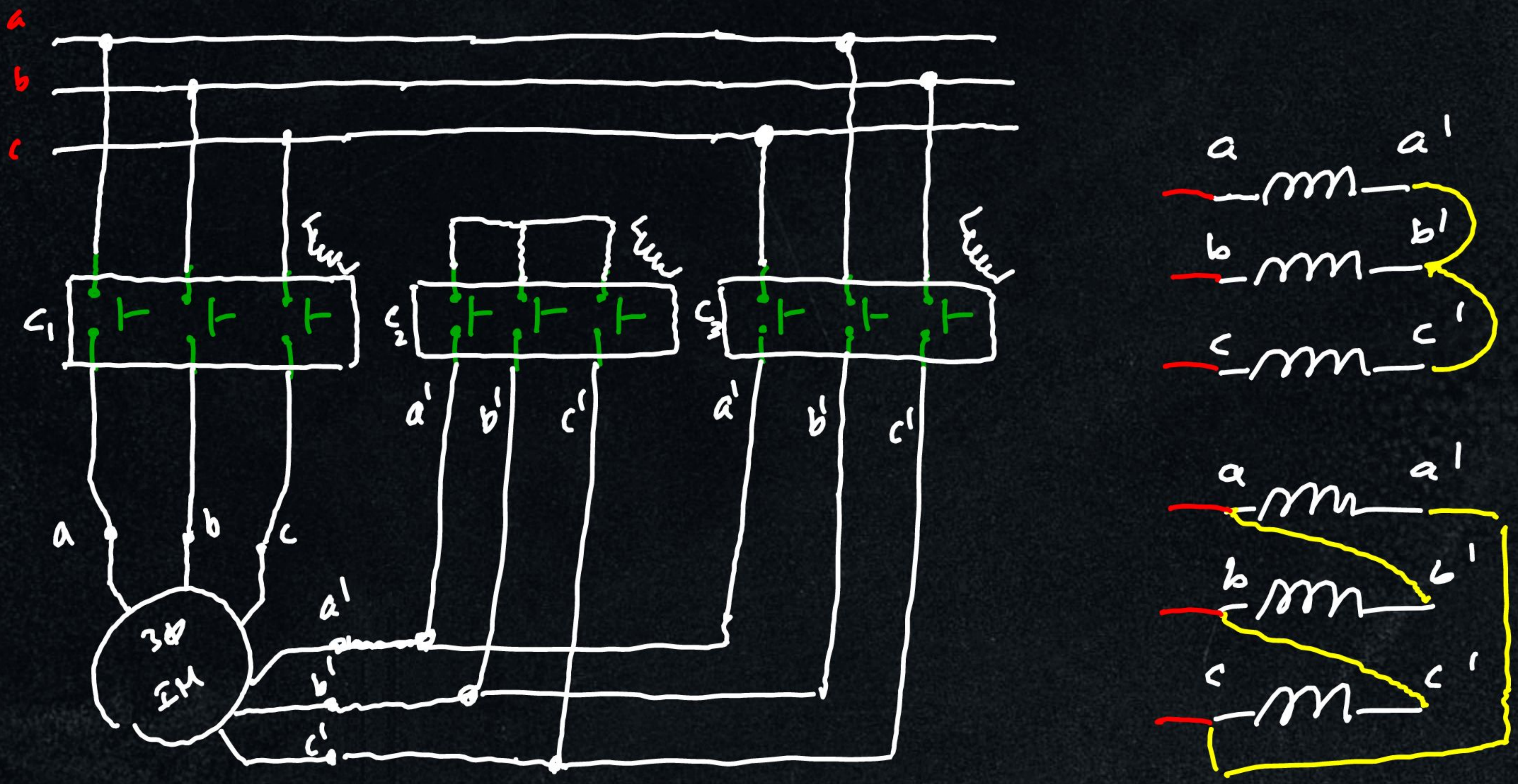
$$I_\Delta = \frac{V}{R}$$



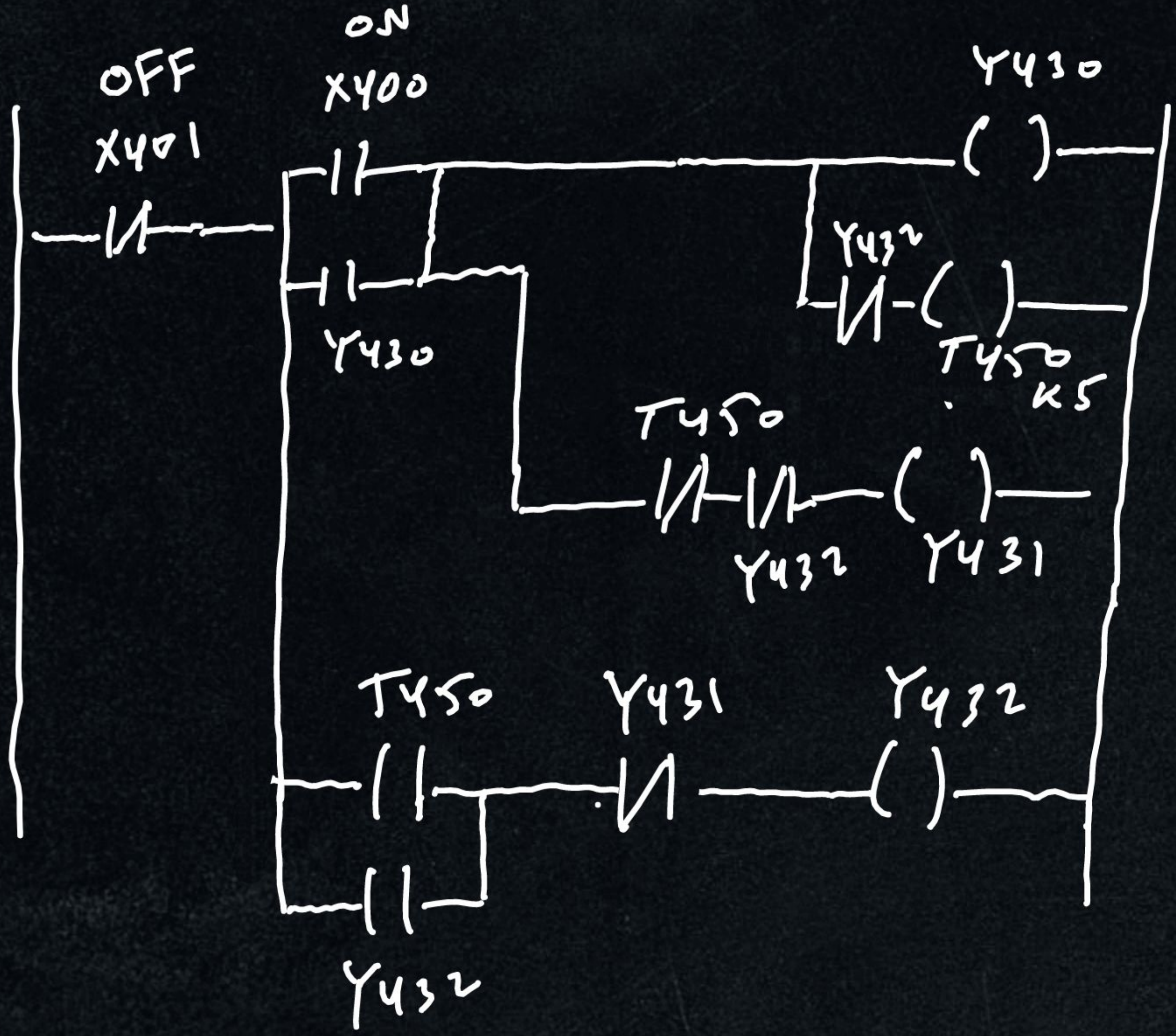


$$\frac{I_Y}{I_\Delta} = \frac{1}{\sqrt{3}} \Rightarrow I_Y = \frac{1}{\sqrt{3}} I_\Delta$$

$$T_{st_Y} = \frac{1}{3} T_{st_\Delta}$$



power circuit

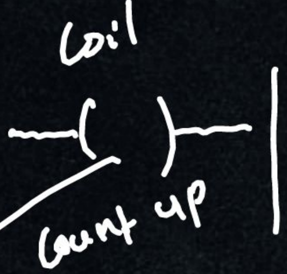


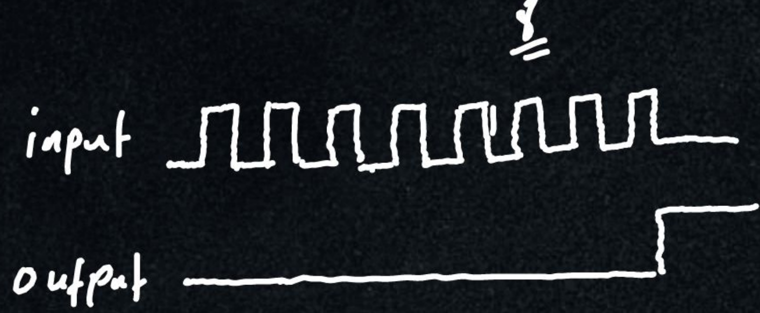
C1 Y430
C2 Y431
C3 Y432

Counters

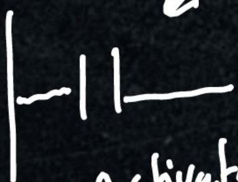
Up-counter → Count from zero up to the preset value

Down-Counter → Count from preset value to zero





50



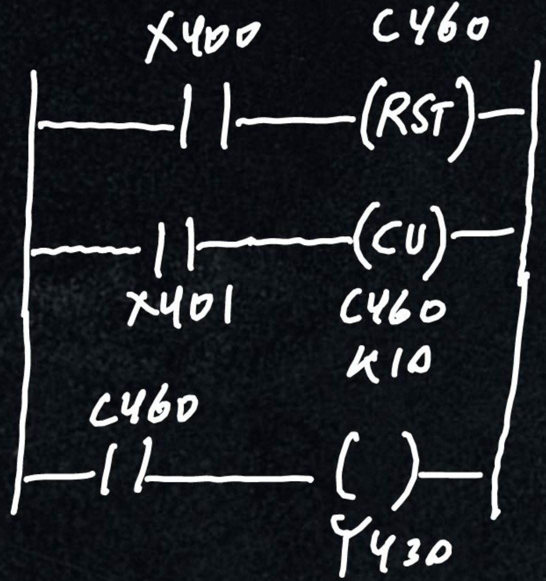
Activated when set count is reached after counting from zero

Addresses

Mitsubishi: C460, C461, ...

Siemens C0, C1, C2, ...

Mitsubishi



LD X400

RST C460

LD X401

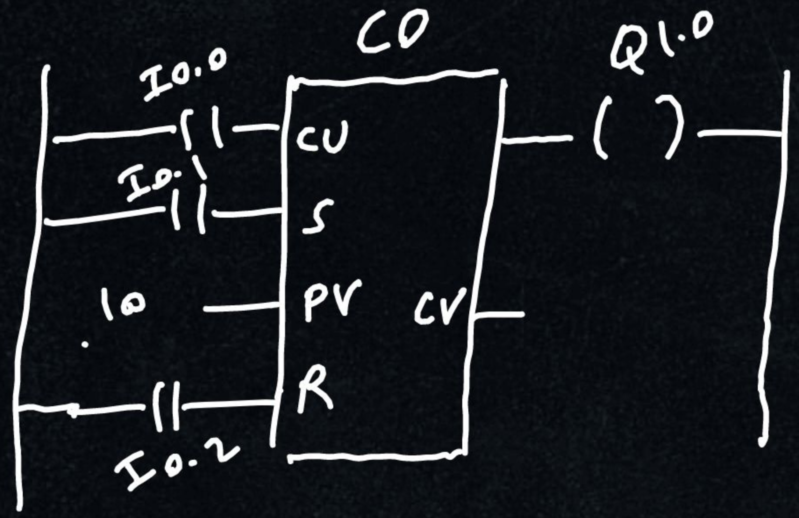
CU C460

K10

LD C460

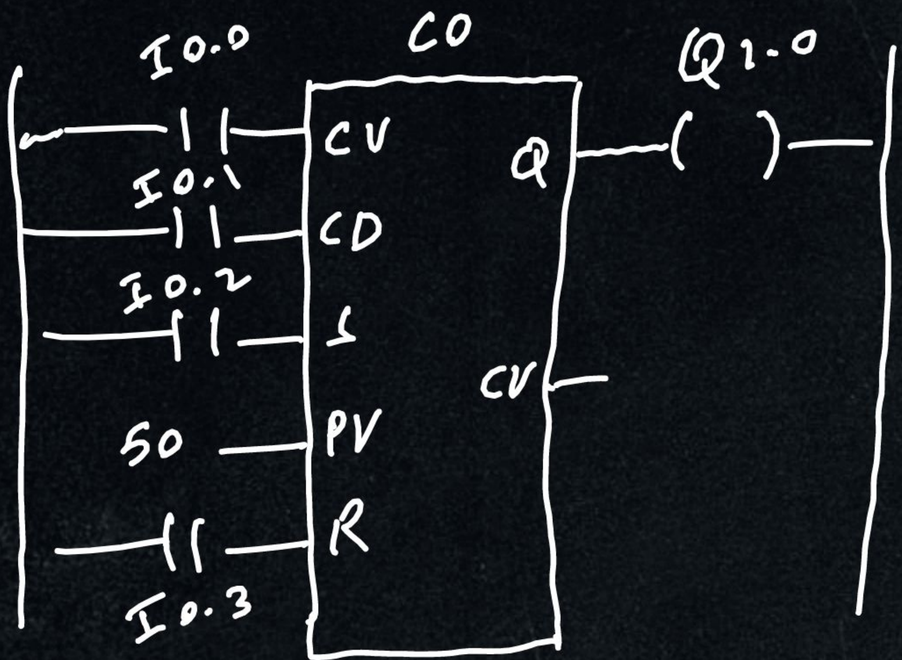
OUT Y430

Siemens



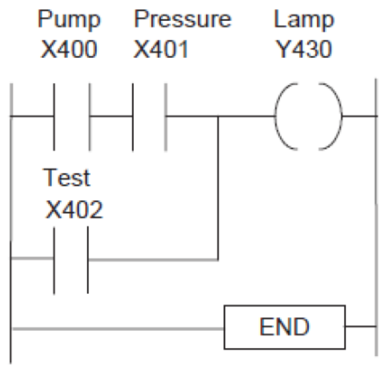
A I 0.0
CU C0
A I 0.1
S C0
LKC 10
A I 0.2
R C0
A C0
= Q1.0

EX :- Design a counter that counts the cars entering a park or leaving it, and gives a message if the park is full.

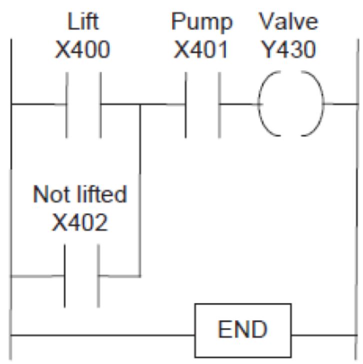


A I 0.0
CU C0
A I 0.1
CD C0
A I 0.2
S C0
L KC 50
A I 0.3
R C0
A C0
= Q 2.0

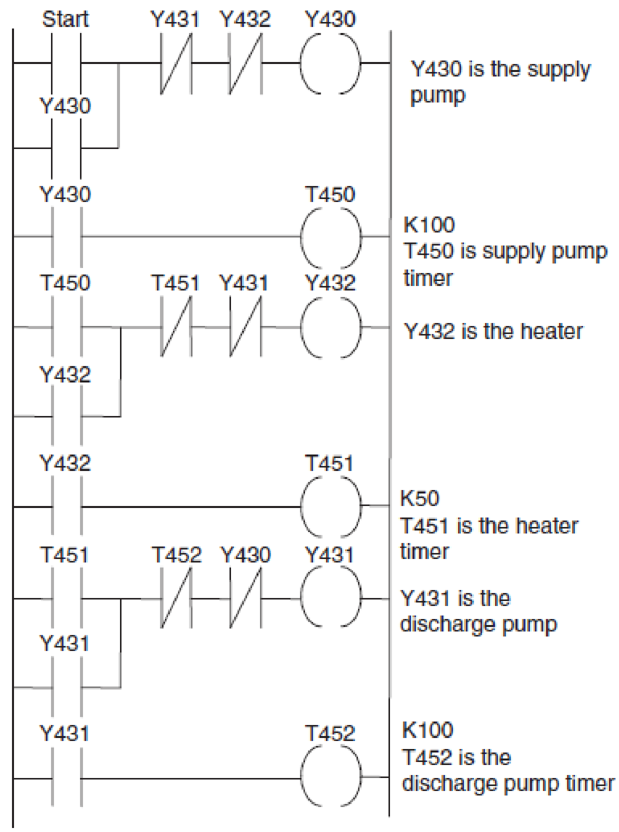
1. Devise a ladder program to switch on a signal lamp if a pump is running and the pressure is satisfactory, or if the lamp test switch is closed.



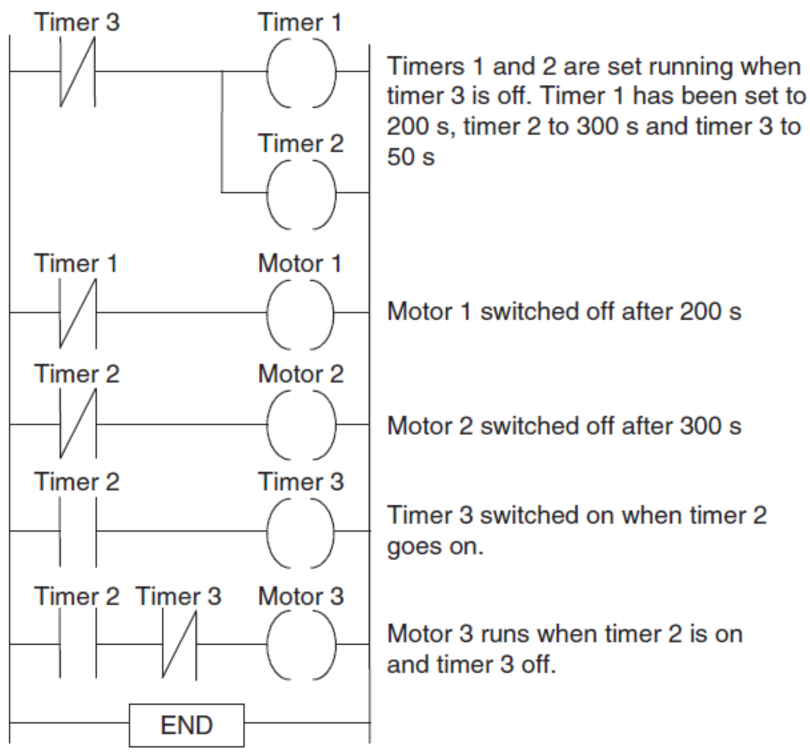
2. Consider a valve which is to be operated to lift a load when a pump is running and either the lift switch is operated or a switch operated indicating that the load has not already been lifted and is at the bottom of its lift channel.



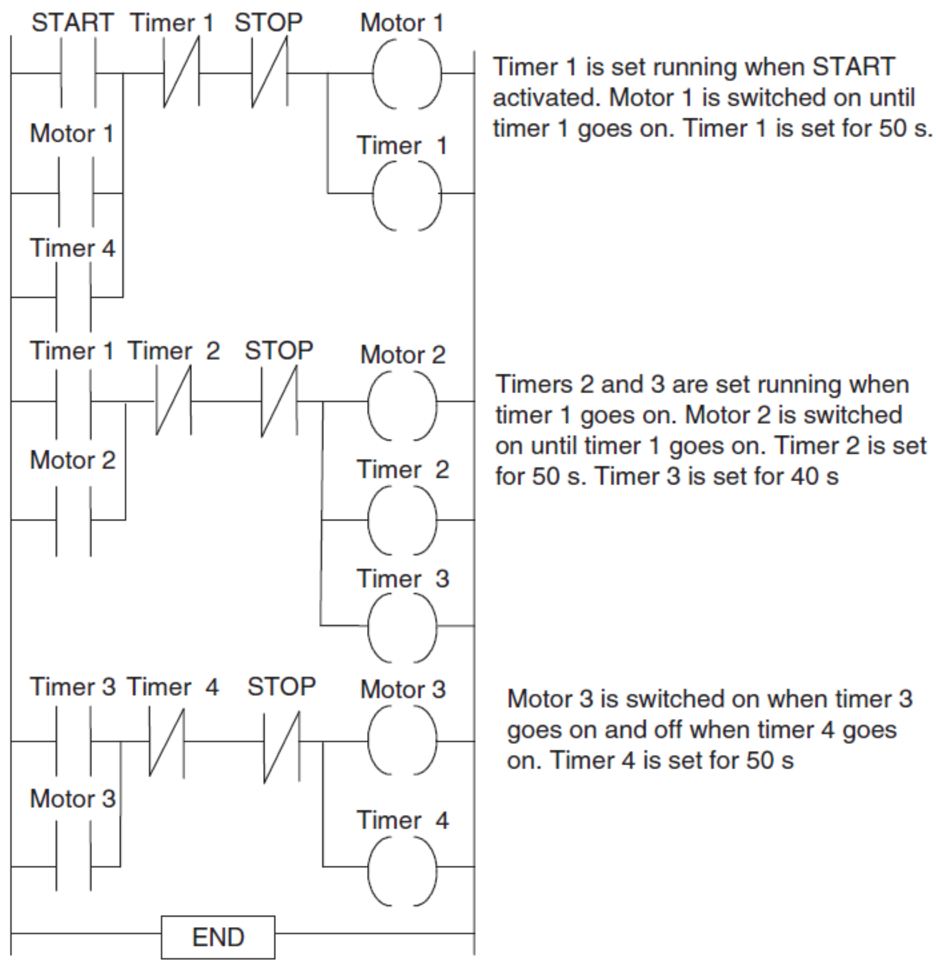
3. This problem is essentially part of the domestic washing-machine program. Devise a ladder program to switch on a pump for 100 s. It is then to be switched off and a heater switched on for 50 s. Then the heater is to be switched off and another pump is to be used to empty the water.



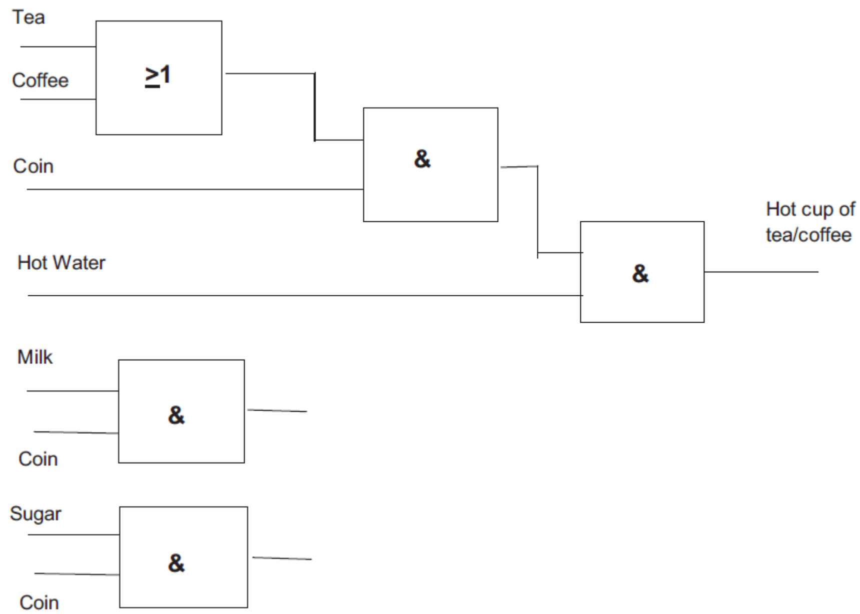
4. Write a ladder program that will switch on two motors when the start switch is operated, then switch off one motor after 200 s and the other motor after a further 100 s. When both motors have been switched off, a third motor is to be switched on for 50 s. The cycle is then to repeat itself unless a stop switch has been activated.

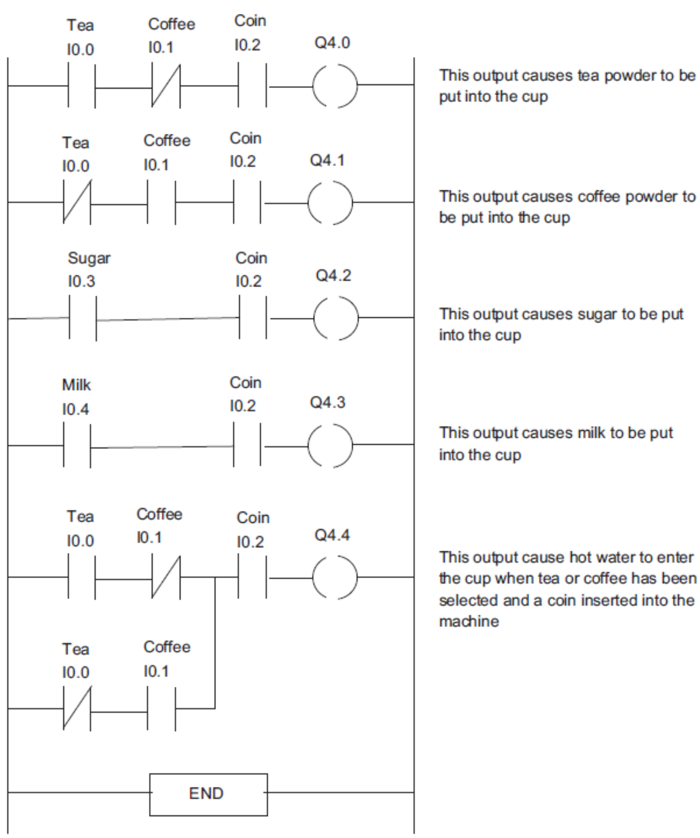


5. Write a ladder program to switch on a motor when the start switch is momentarily activated, with the motor remaining on for 50 s. At the end of that time a second motor is to be switched on for a further 50 s. A third motor is to be switched on 10 s before the second motor switches off and is to remain on for 50 s. The cycle is then to repeat itself unless a stop switch has been activated.



6. Write a FBD and ladder programs for drinks machine that allows the selection of tea or coffee, milk or no milk, sugar or no sugar, and will supply the required hot drink on the insertion of a coin.





7. Consider the problem of the control of a machine which is required to direct 6 tins along one path for packaging in a box and then 12 tins along another path for packaging in another box. A deflector plate might be controlled by a photocell sensor which gives an output every time a tin passes it. Thus the number of pulses from the sensor has to be counted and used to control the deflector.