Relay + Programmable Logic Controllers

1) RLC

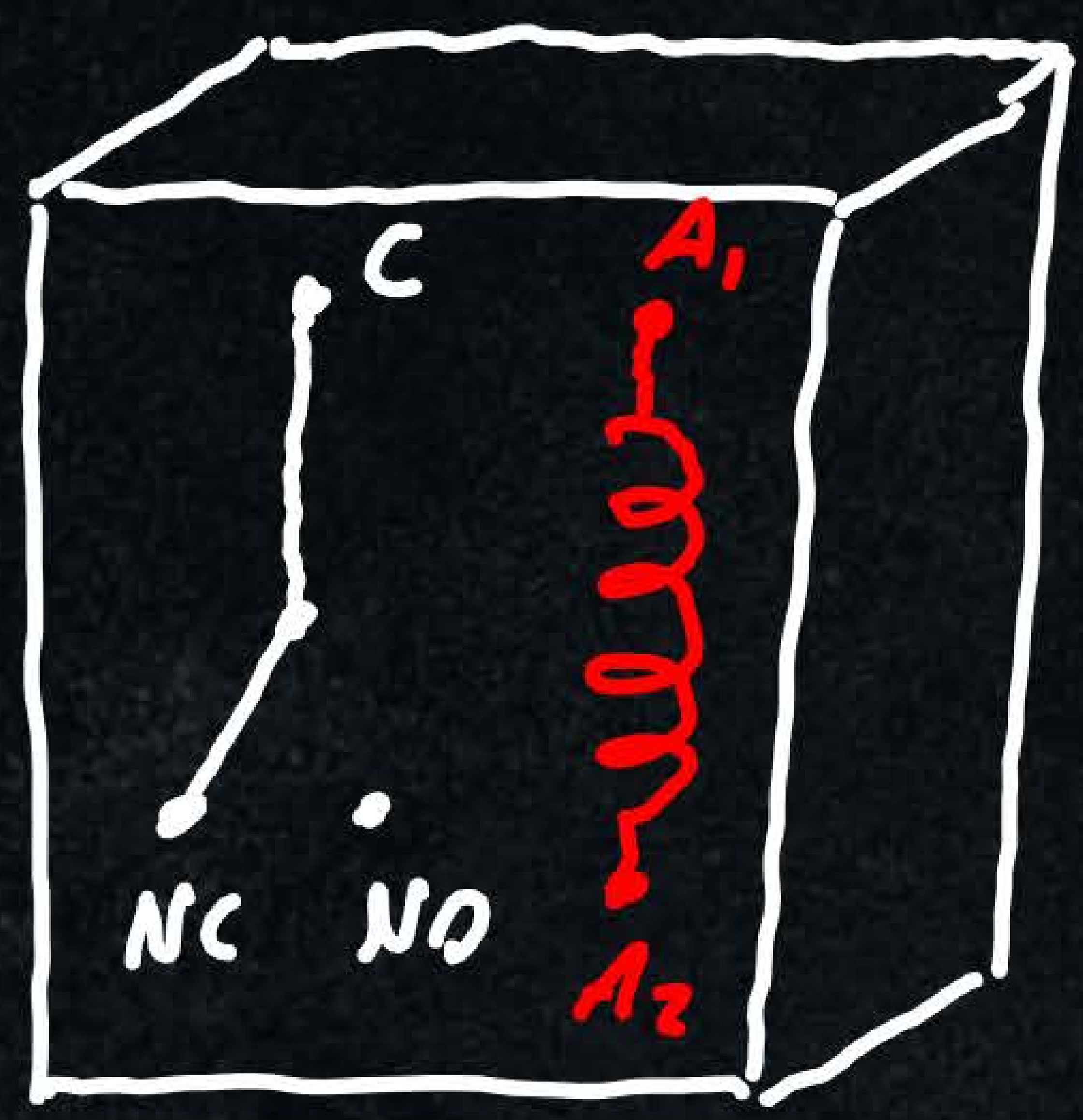
2) PLC

① RLC

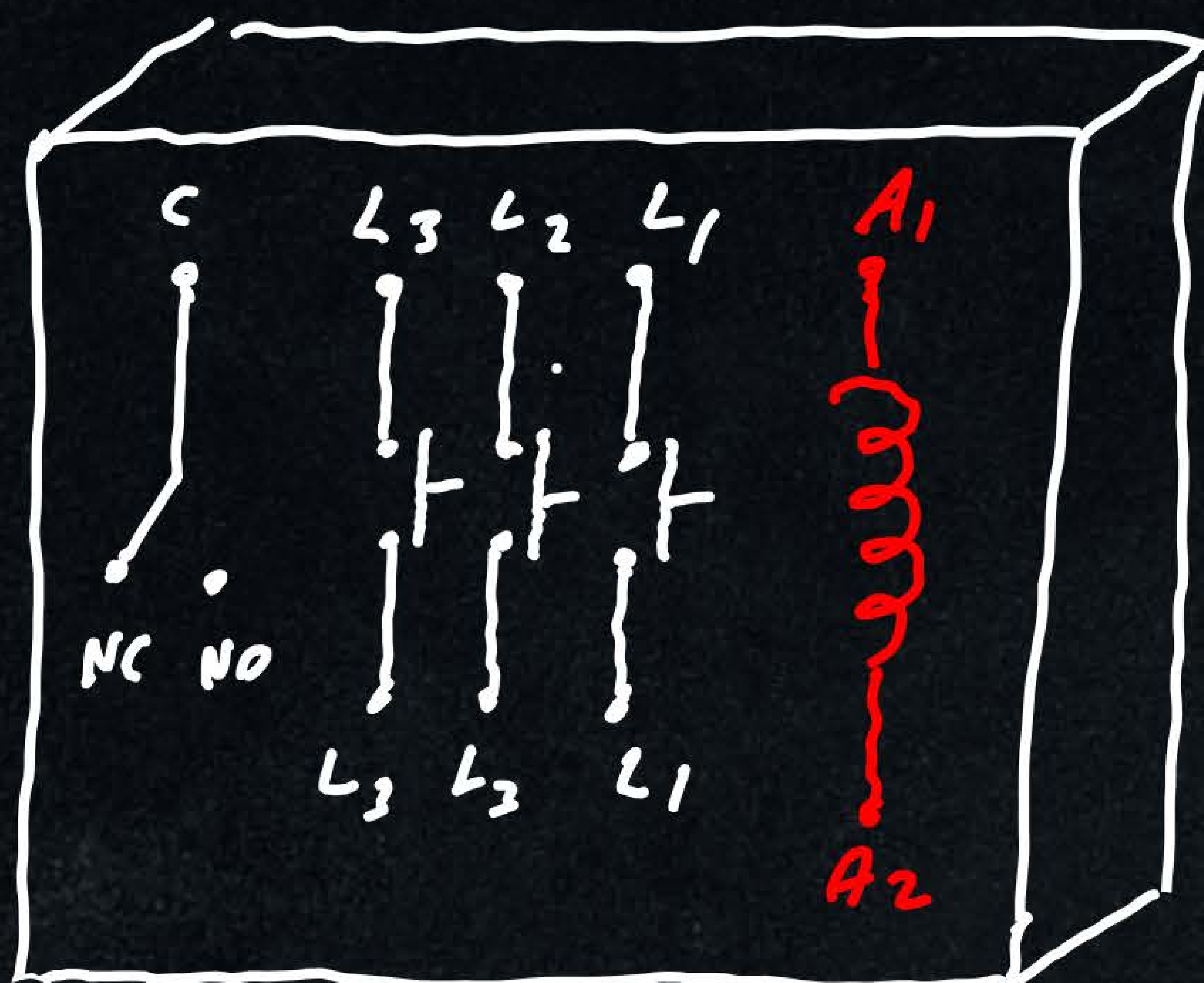
- prior to the widespread use of PLCs in process control and automation, hardwired relay control systems were used.
- Control of motors such as starting, stopping, reversing the direction, sequence starting, etc can be achieved by designing suitable control circuits using relays, contactors, push buttons,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

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

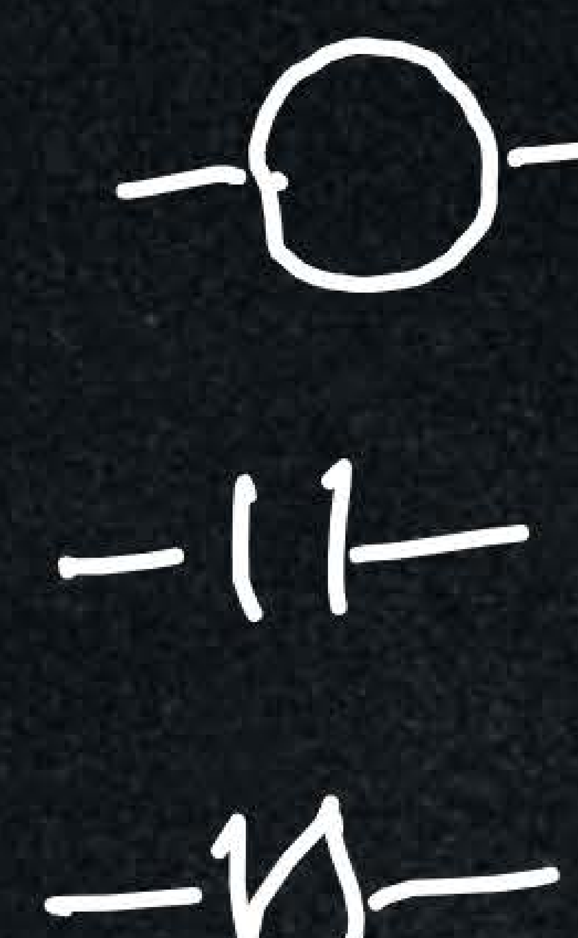
→ Contacts can be either normally open or normally closed. (Auxiliary contacts)



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



Relay symbols



Relay coil

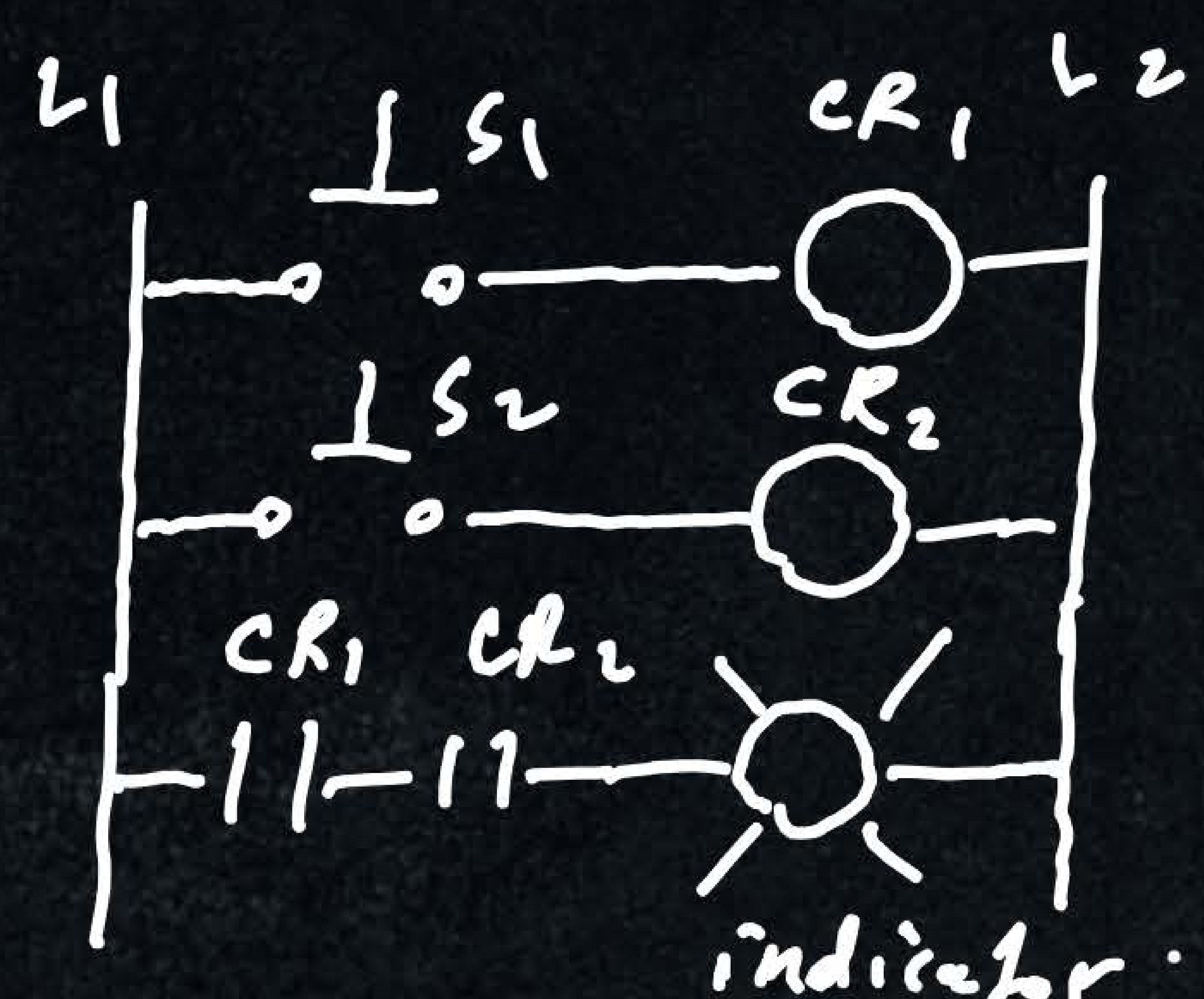
Normally open contact

Normally closed contact

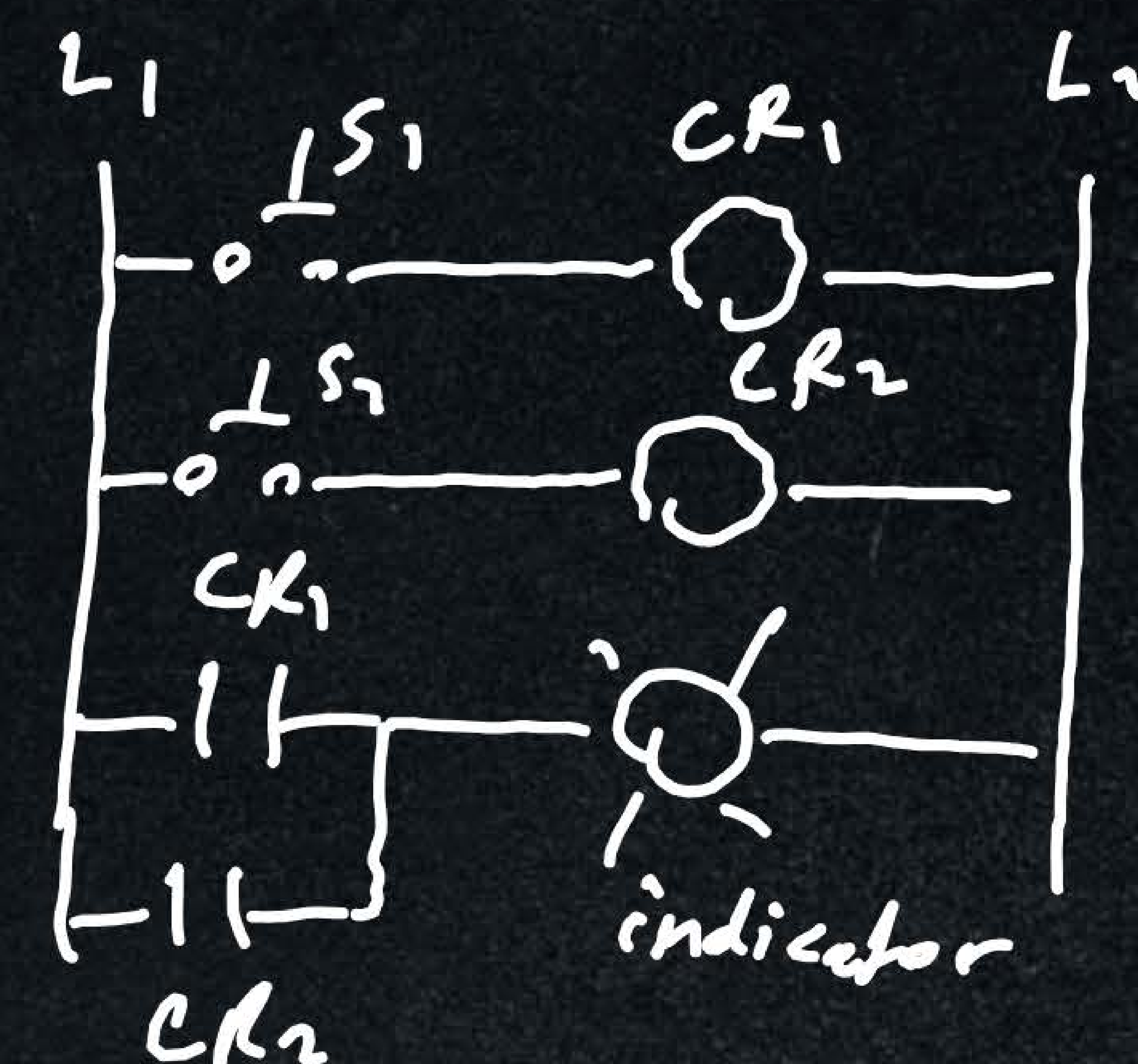
Line Diagram

Examples

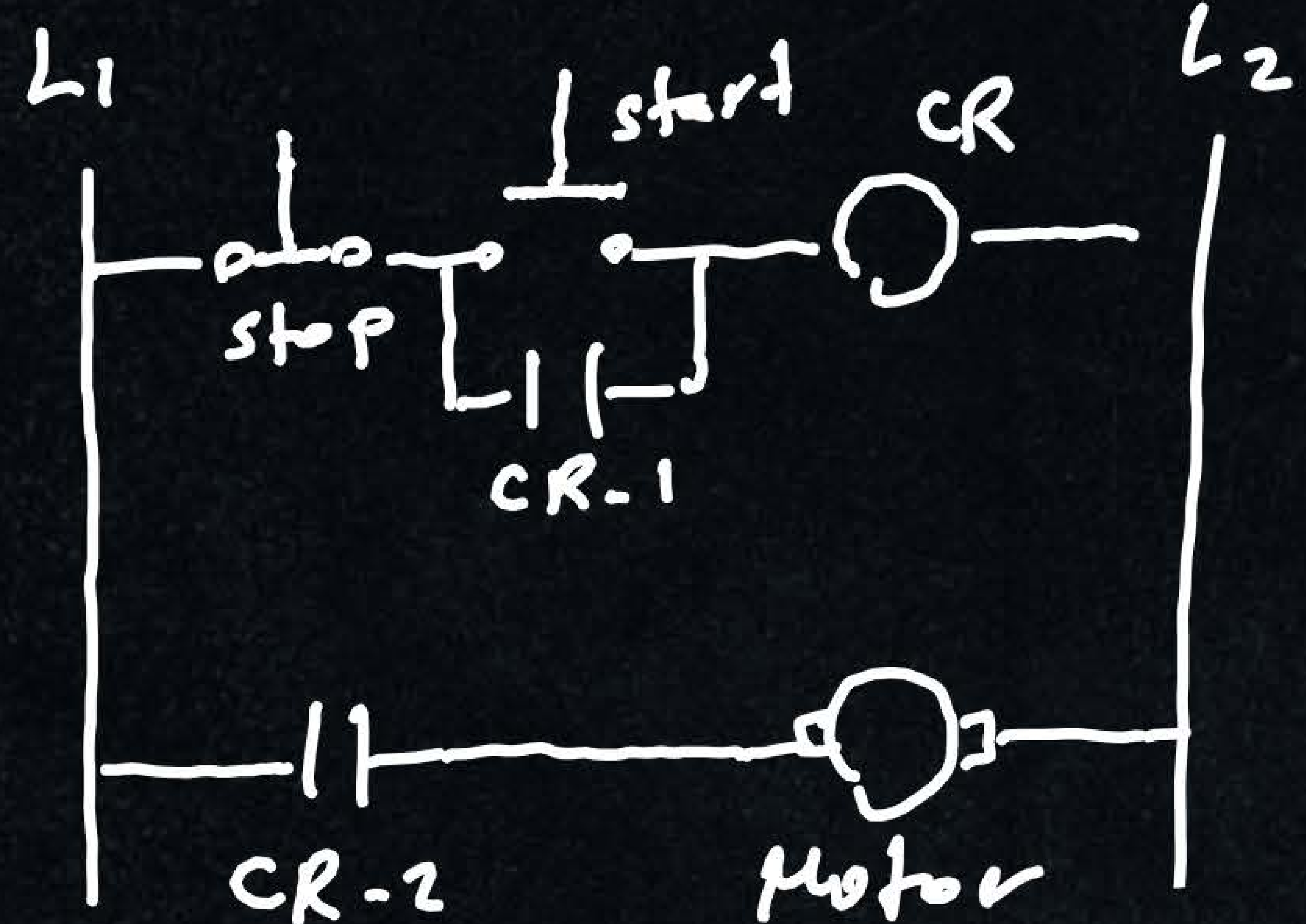
① AND logic



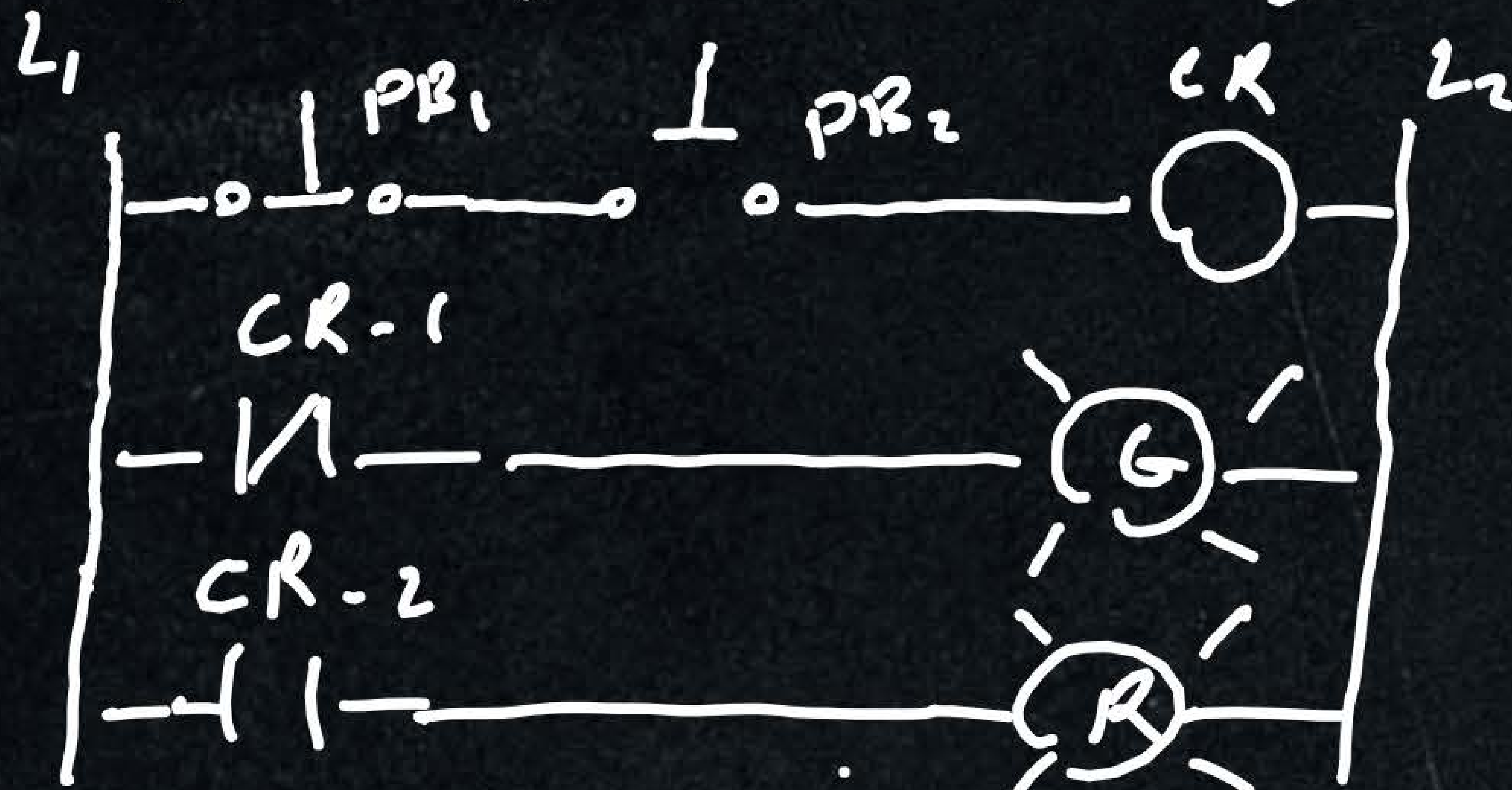
② OR logic



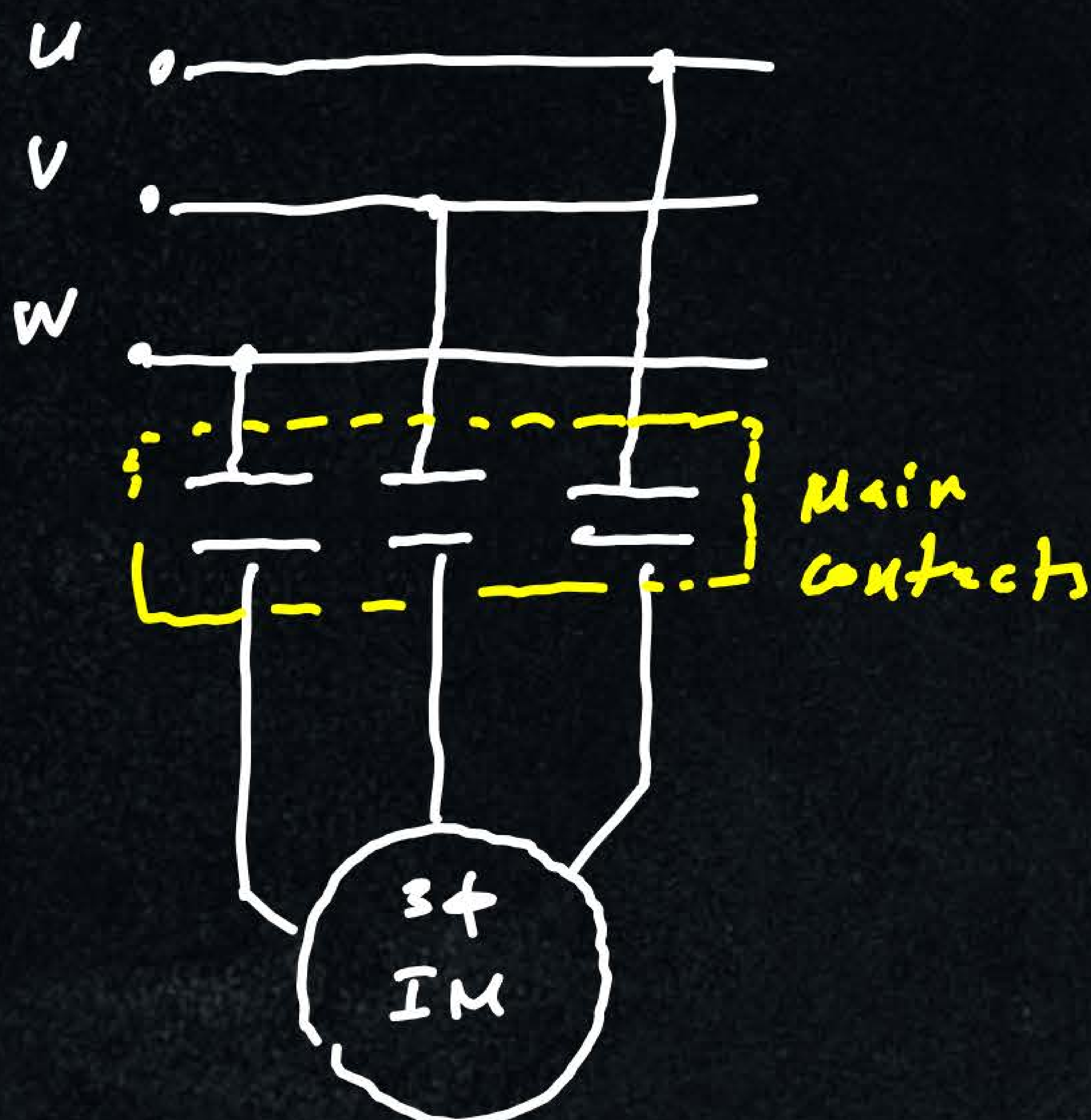
3) start / stop DC Motor

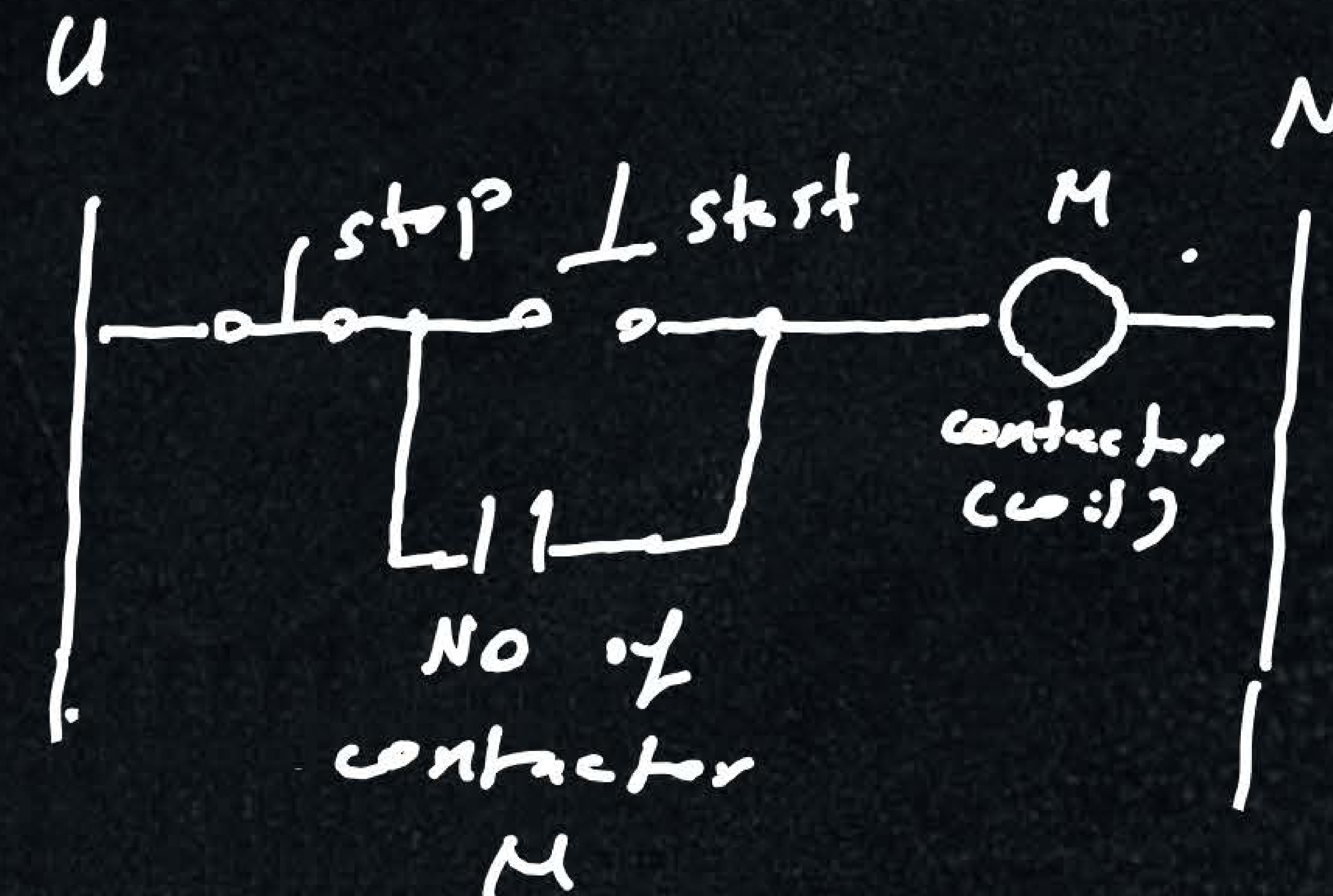


4) Relay controlling two pilot lights.

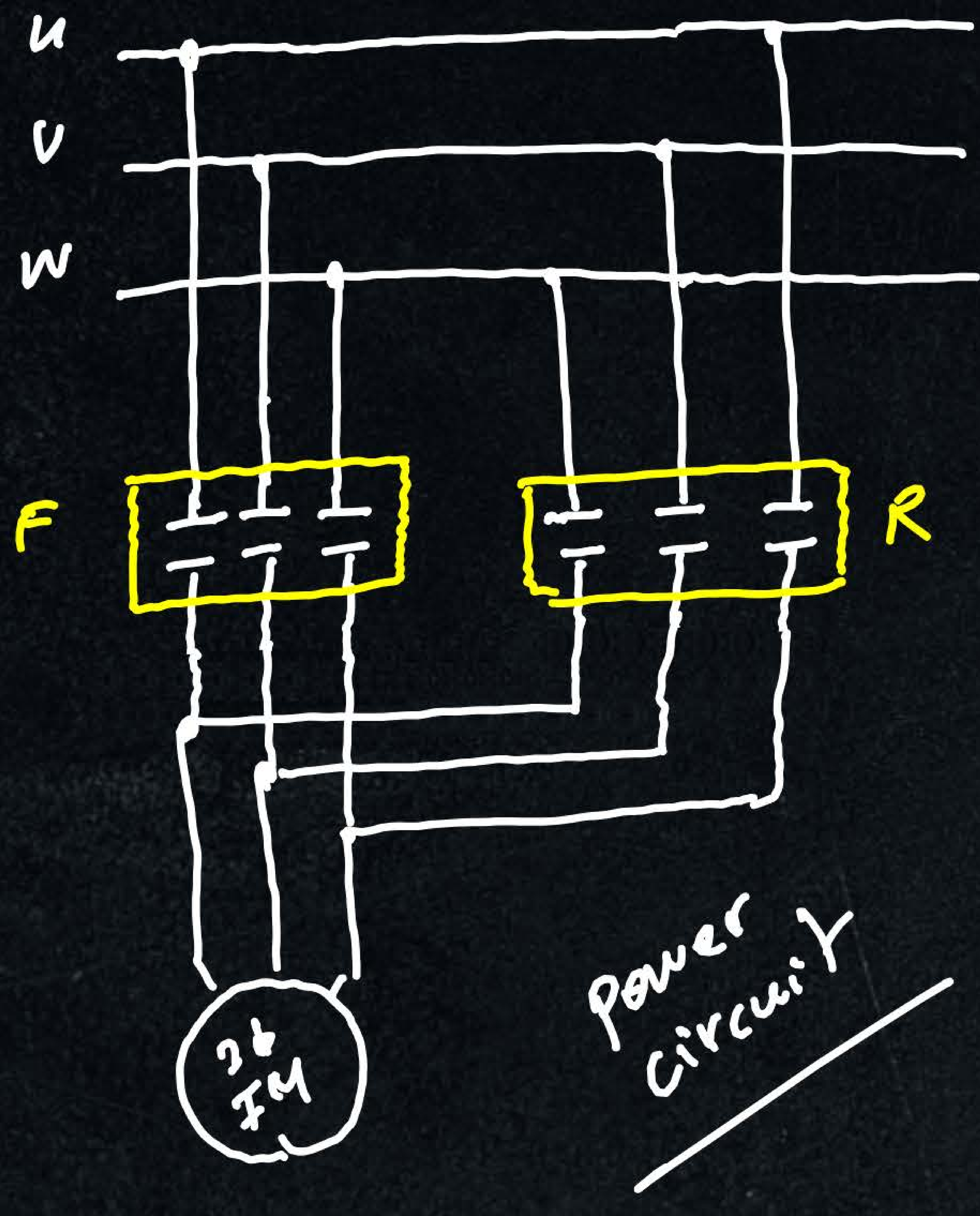


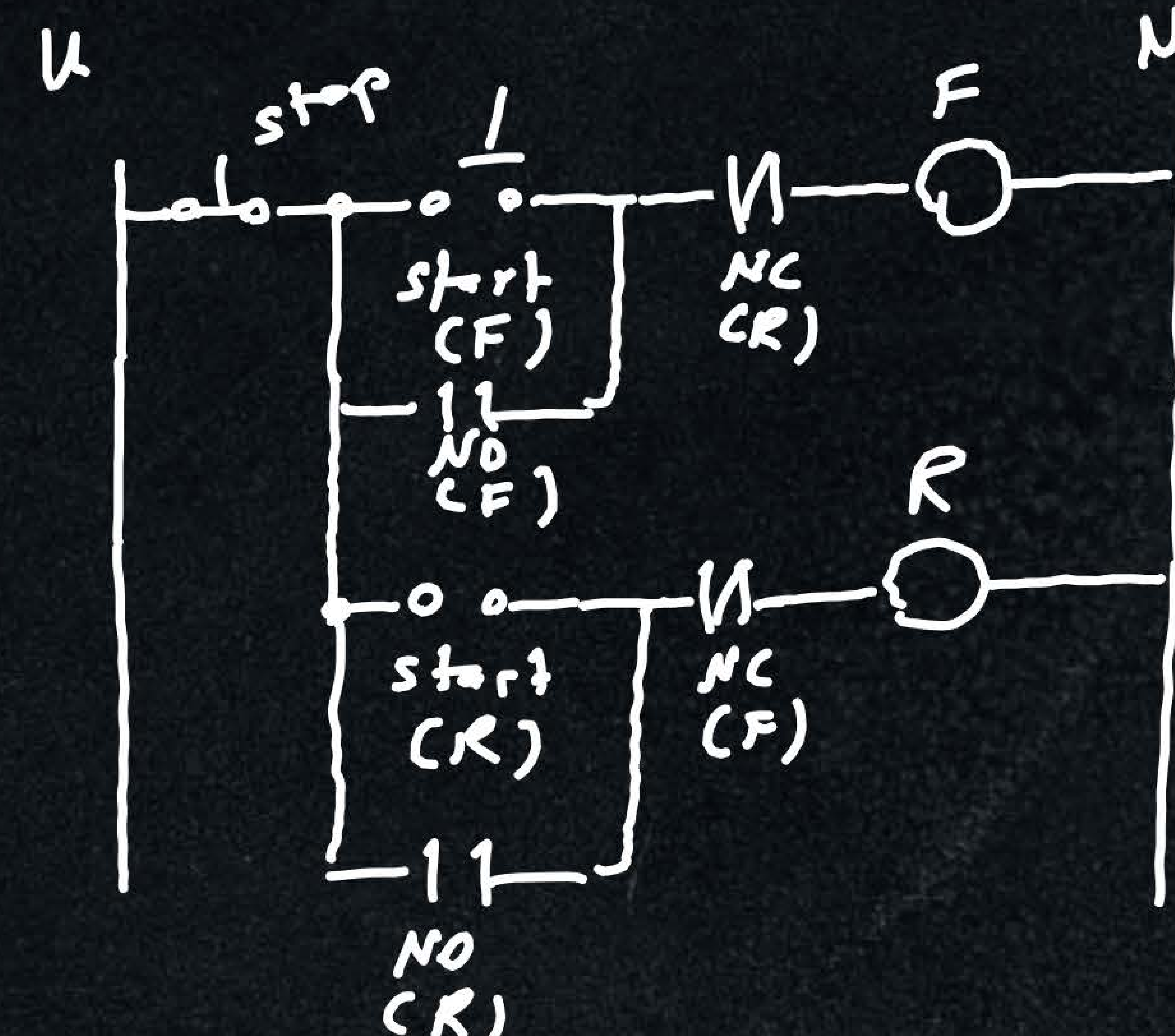
5) Start / Stop of 3-φ Motor



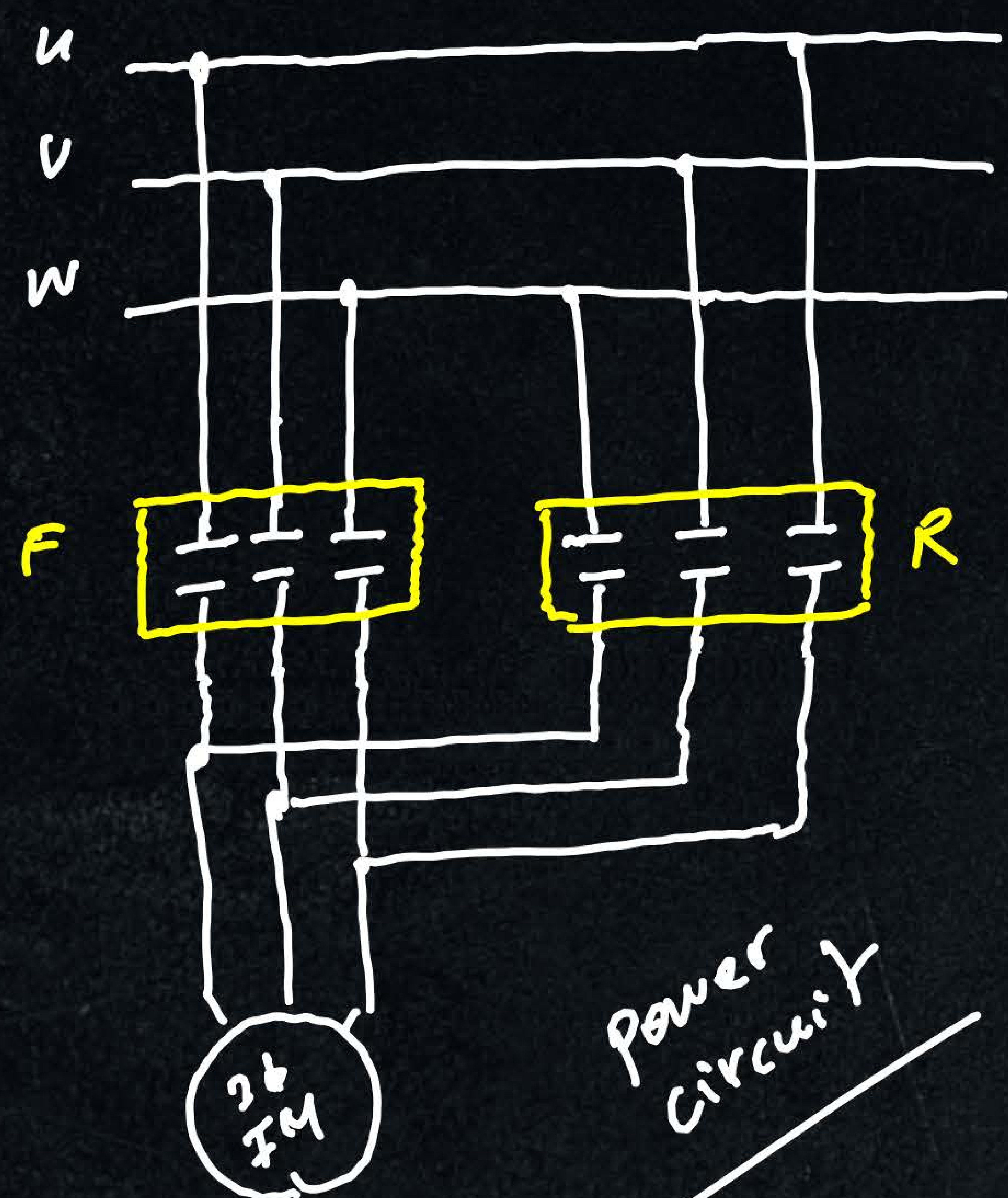


6) Reversing the direction of 3φ motor + start/stop

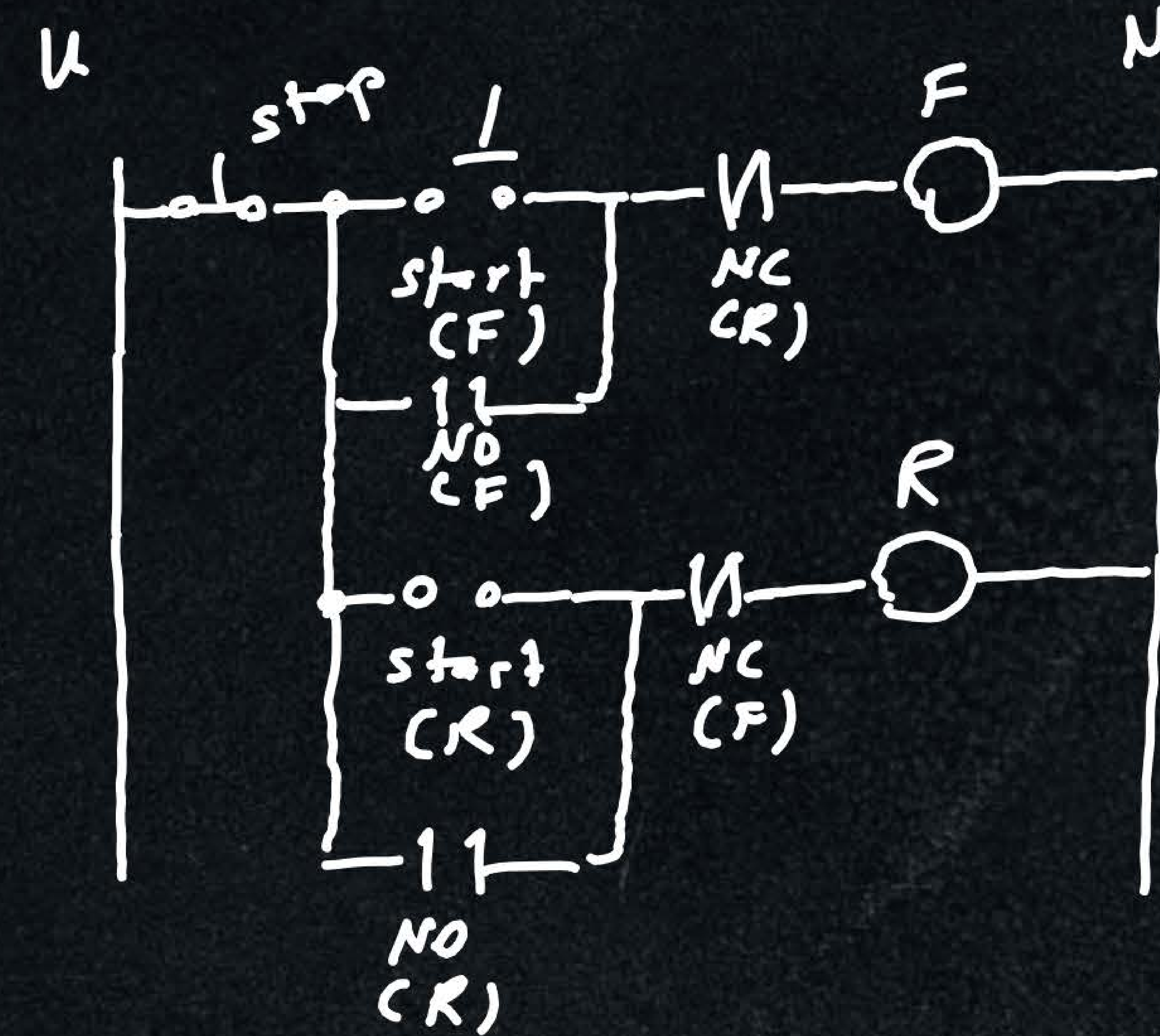




6) Reversing the direction of 3φ motor + start/stop



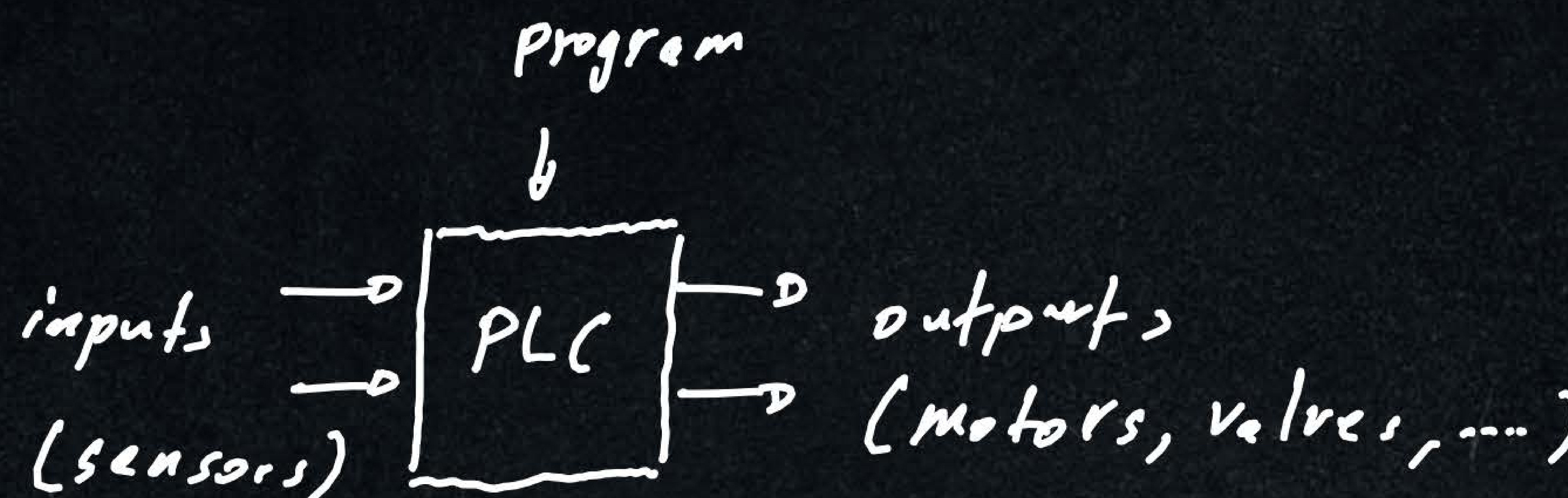
power circuit



control circuit

② PLC

- It is a special from of microprocessor-based controller that uses memory to store instructions and to implement functions such as logic, sequencing, counting, and arthmatic in order to control machines and processes.

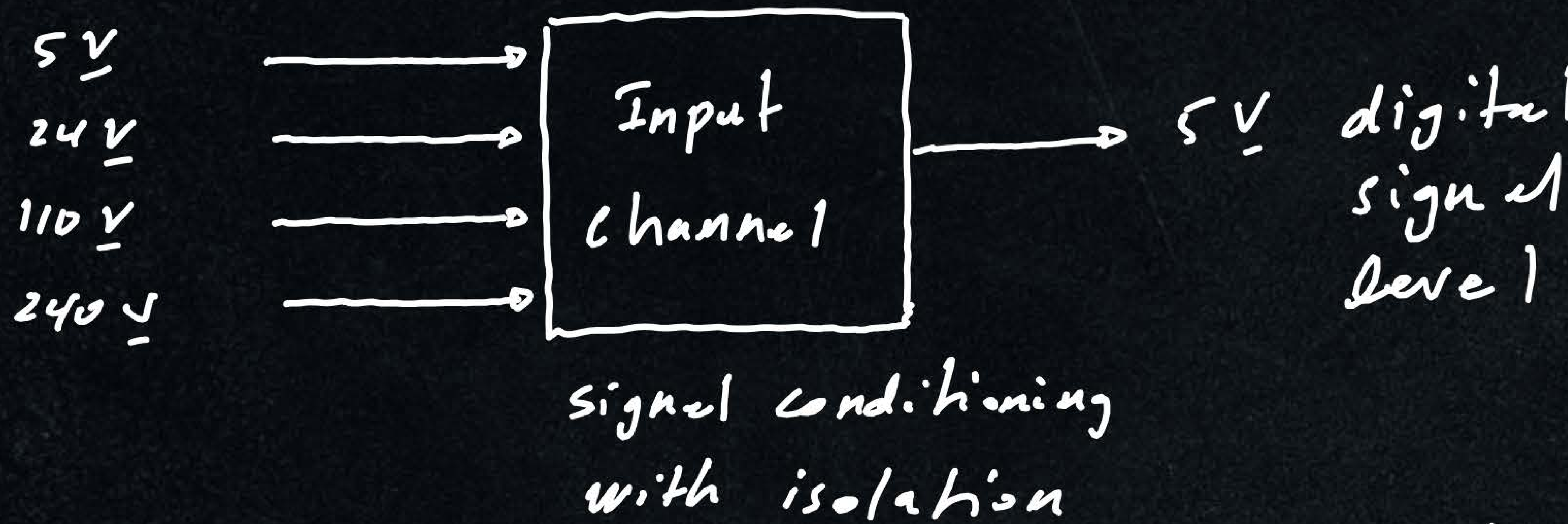


- Advantages (compared to RLC)

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

Block Diagram

Input / output levels



PLCs :

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

programming Languages :

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

I/O Addresses

* The PLC has to be able to identify each input and output. It does this by allocating addresses to each input and 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 notation

  16 outputs Y430 - Y447 in octal notation

* 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

- X → I: input, Q: output
- XX → byte number
- X → bit number

<u>EX:</u>

I 0.1 input at bit 1 in byte 0

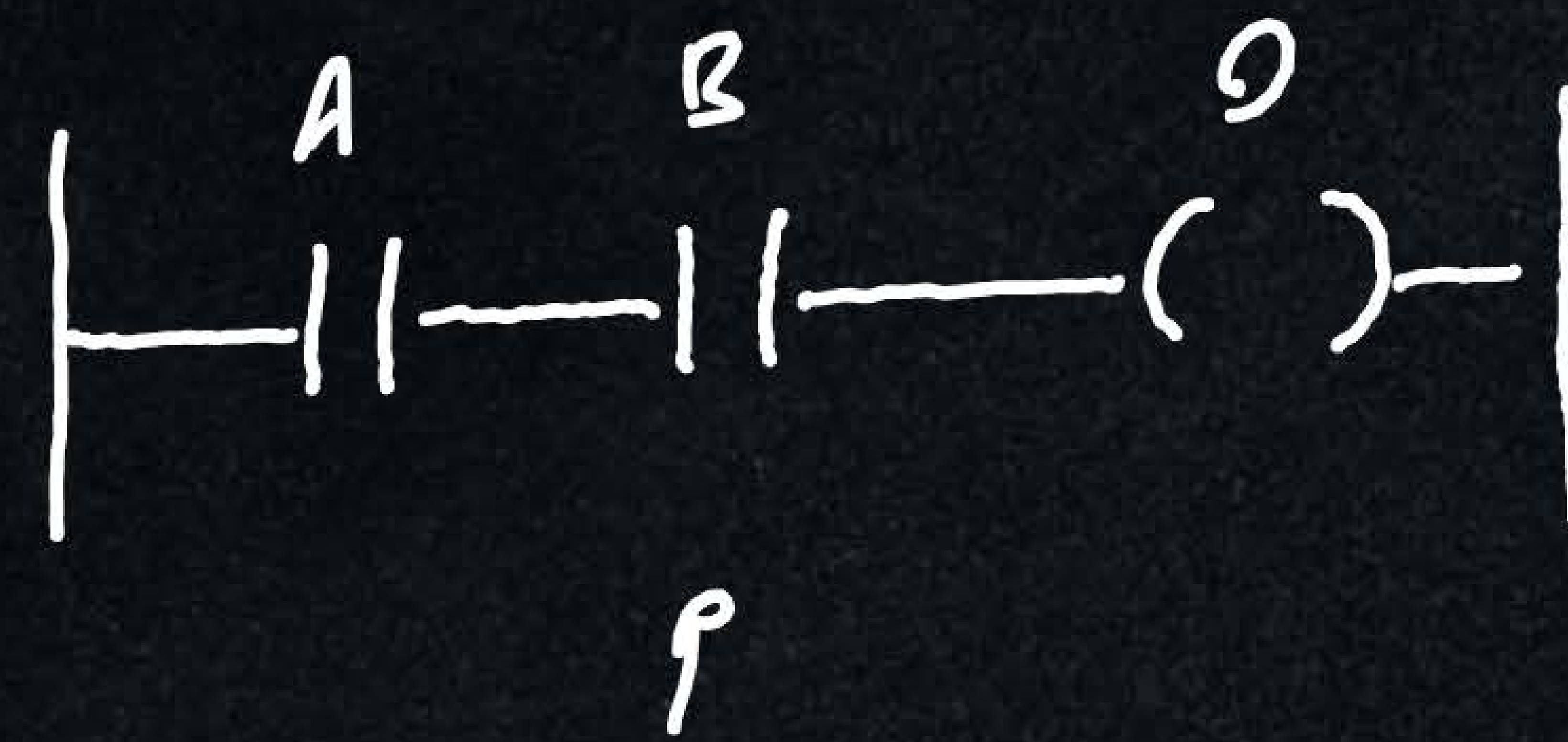
Q 2.0 output at bit 0 in byte 2

Logic Functions (LAD)

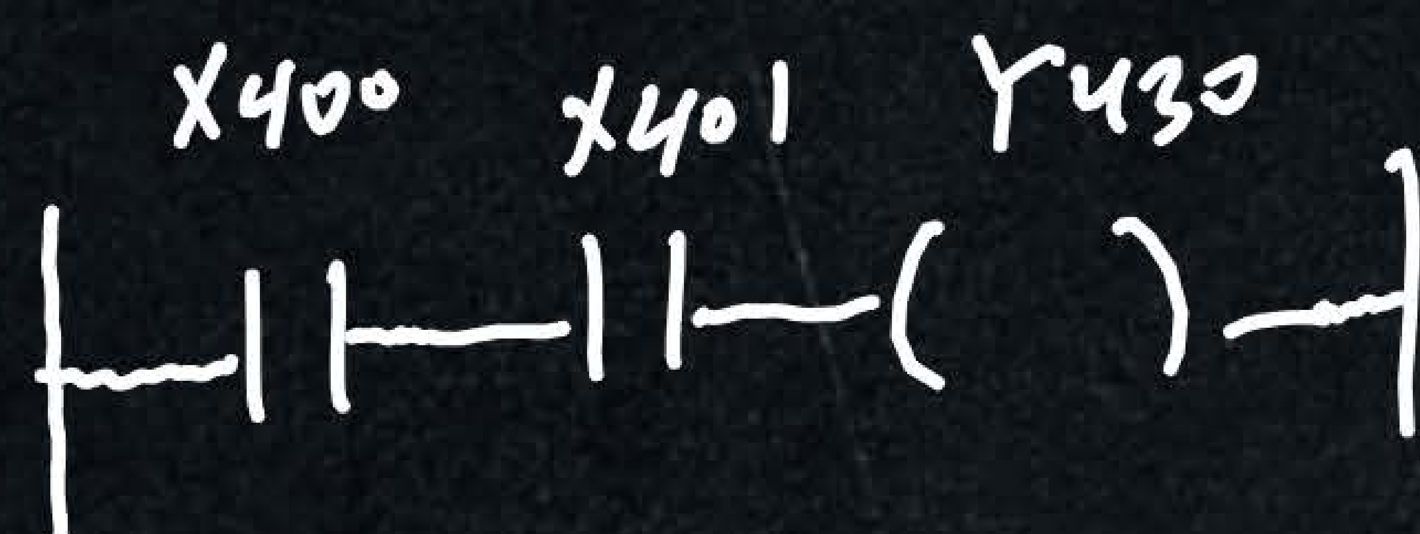
1) AND

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

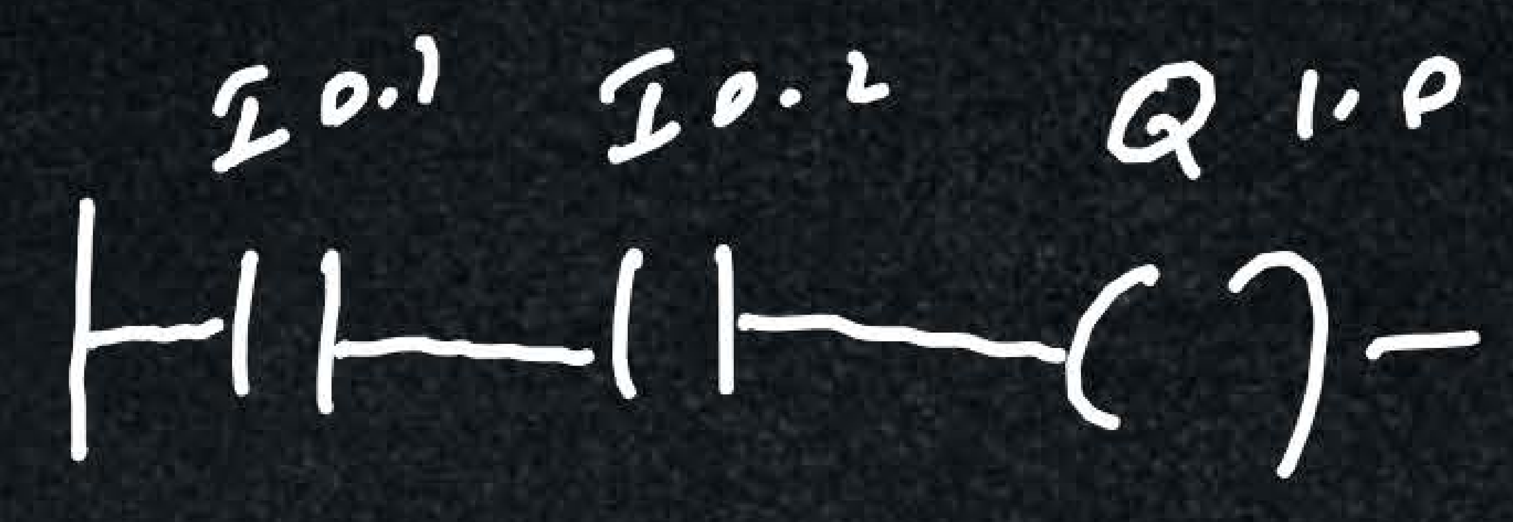
$O = AB$



LAD



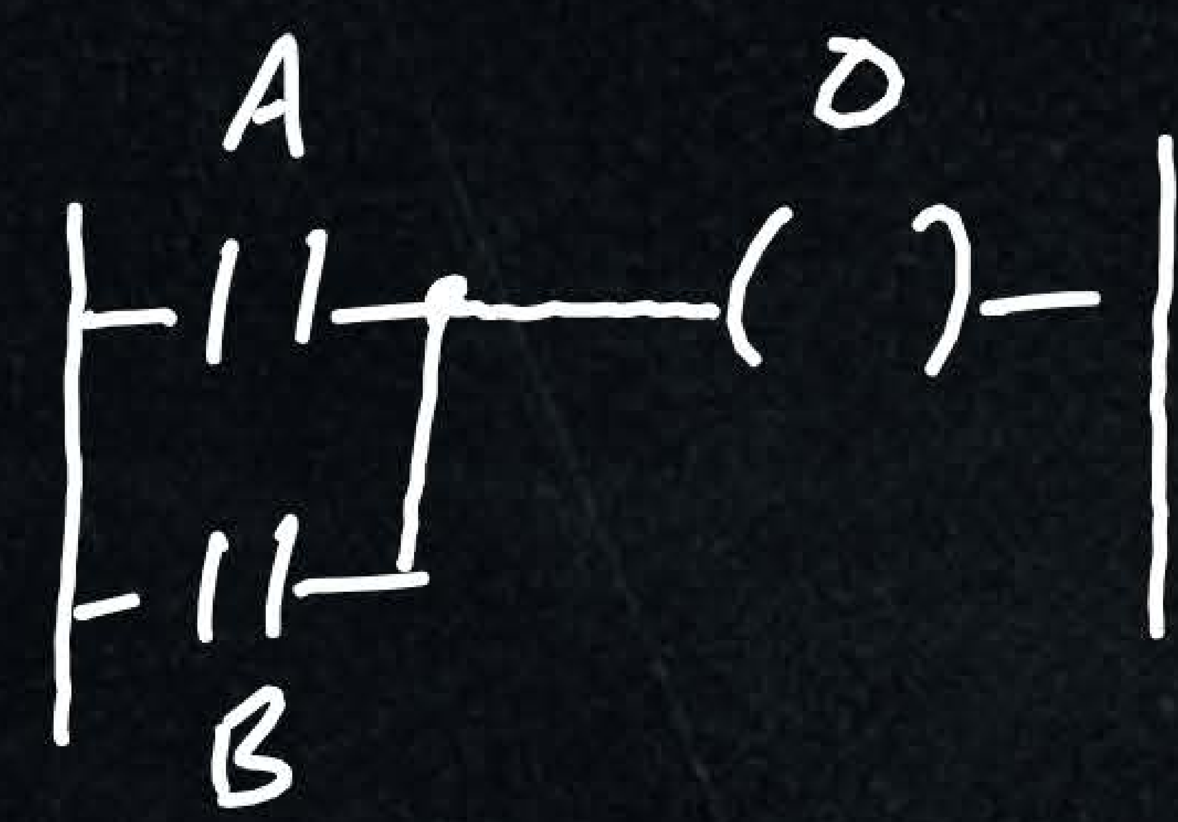
Mitsubishi



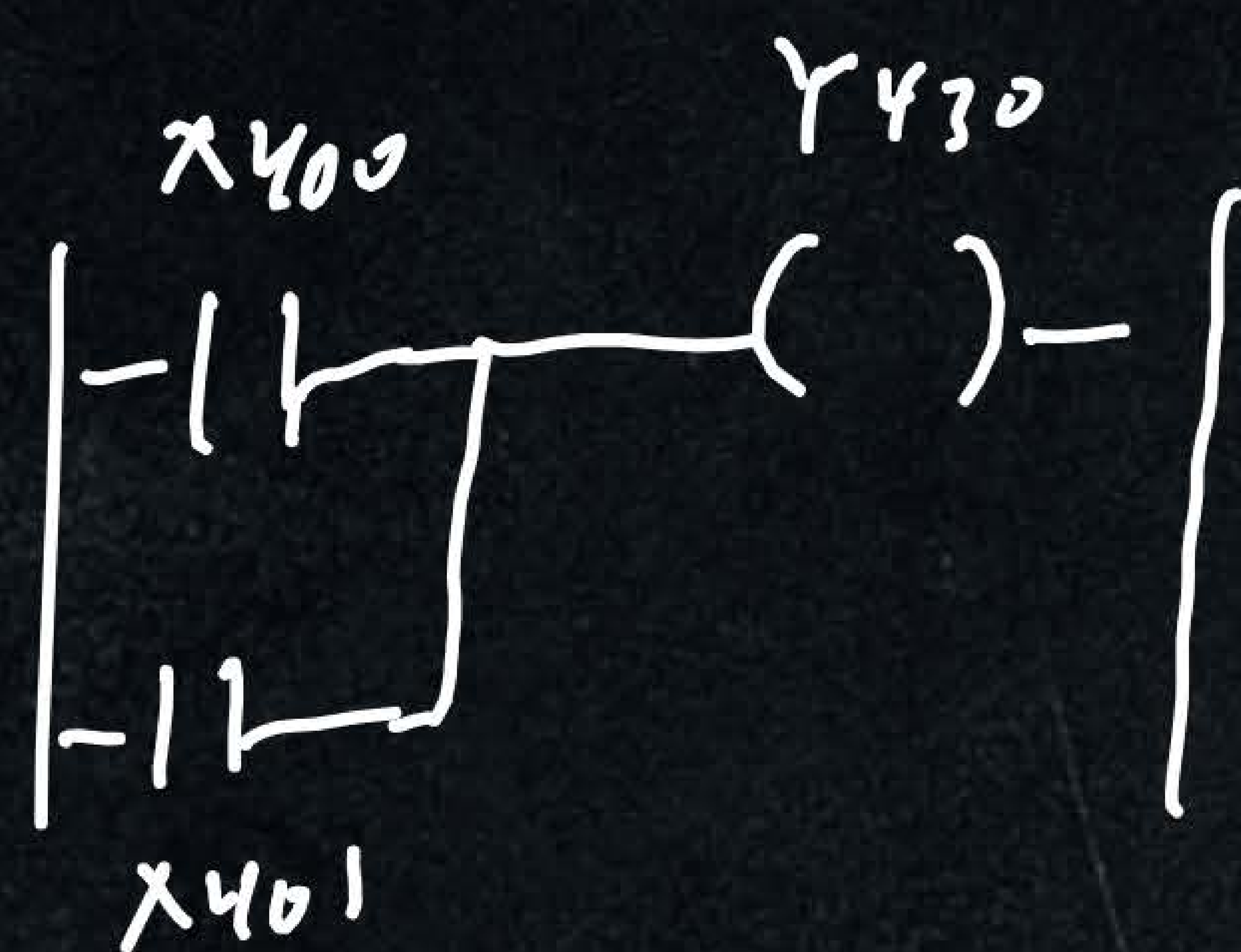
Siemens

2) OR

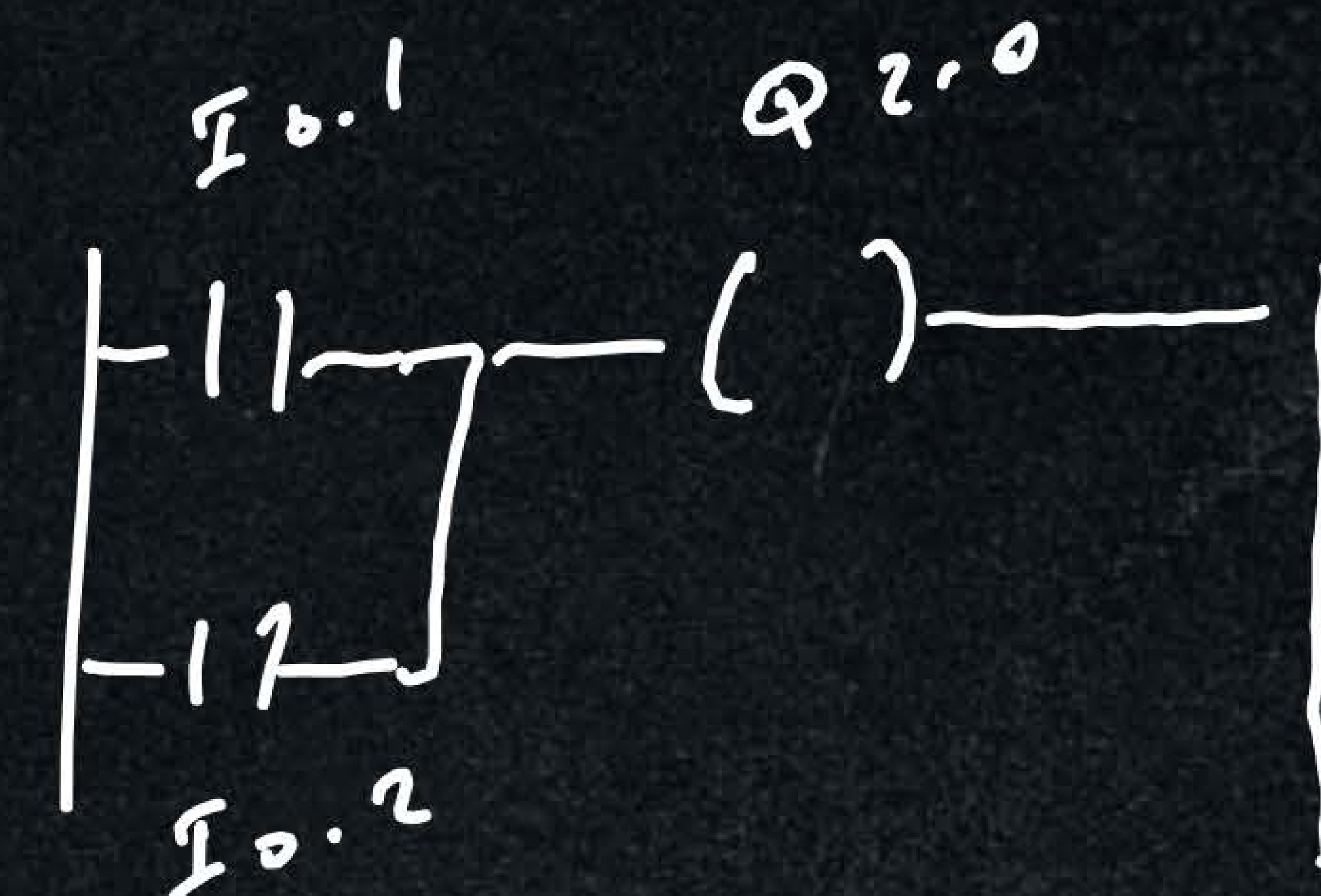
A	B	O
0	0	0
0	1	1
1	0	1
1	1	1



O = A + 13



Mitsubishi

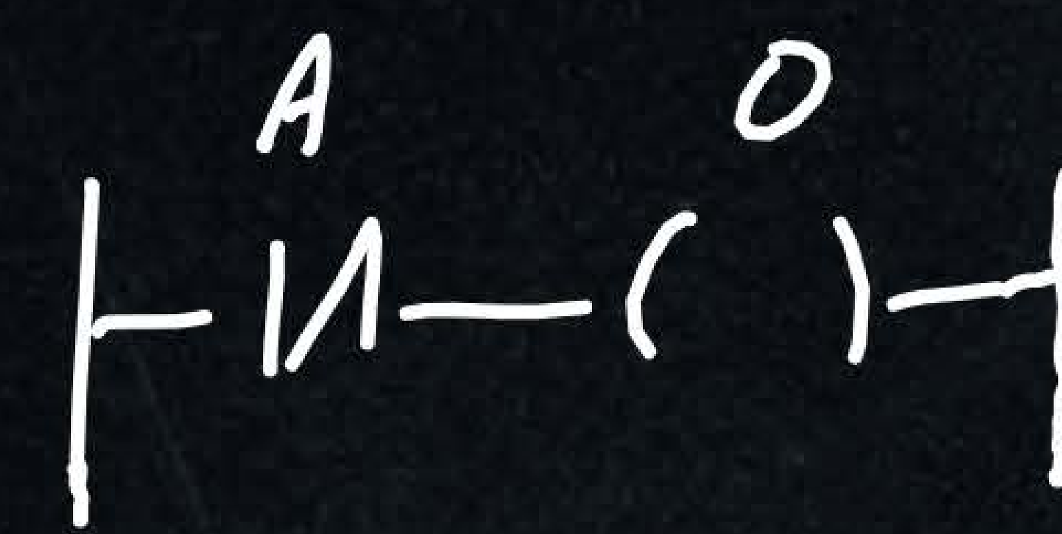


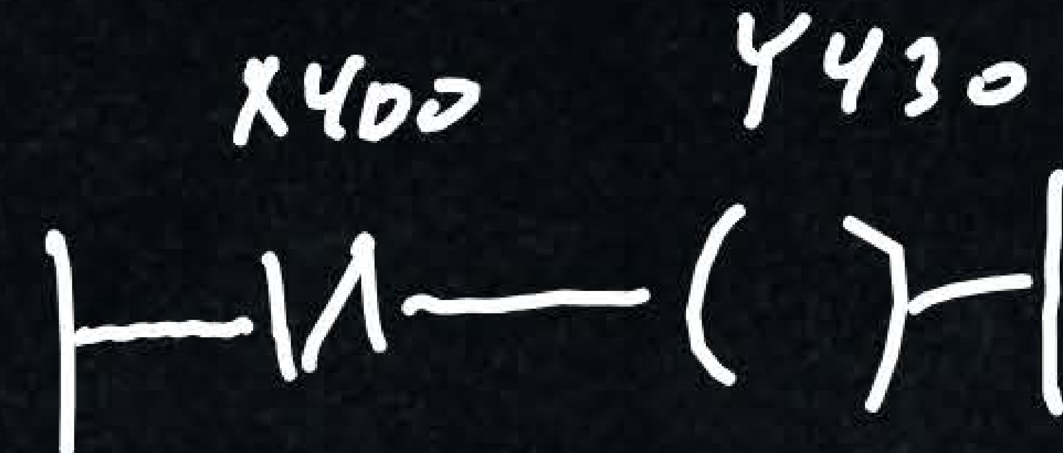
Siemens

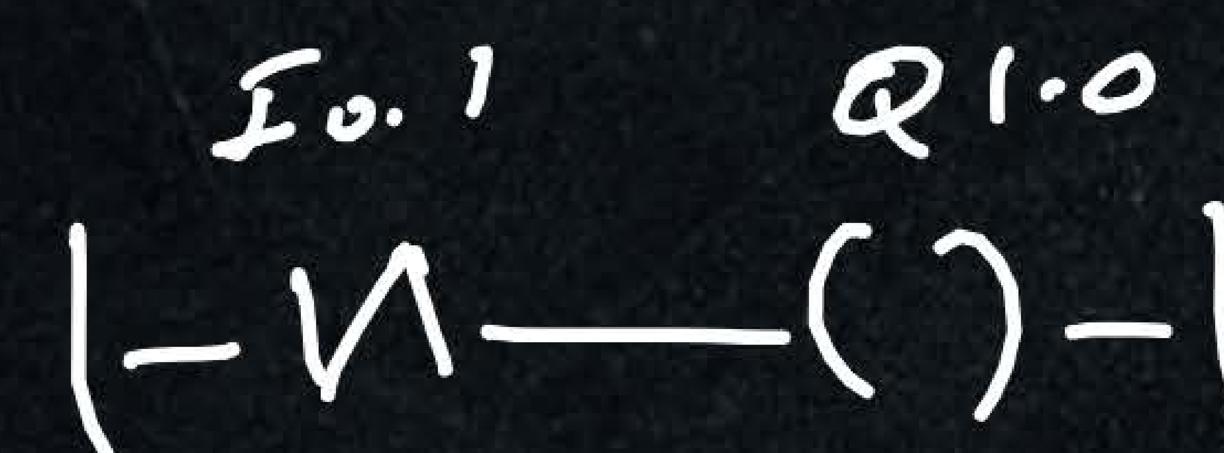
3) NOT

A	O
0	1
1	0

$O = A'$



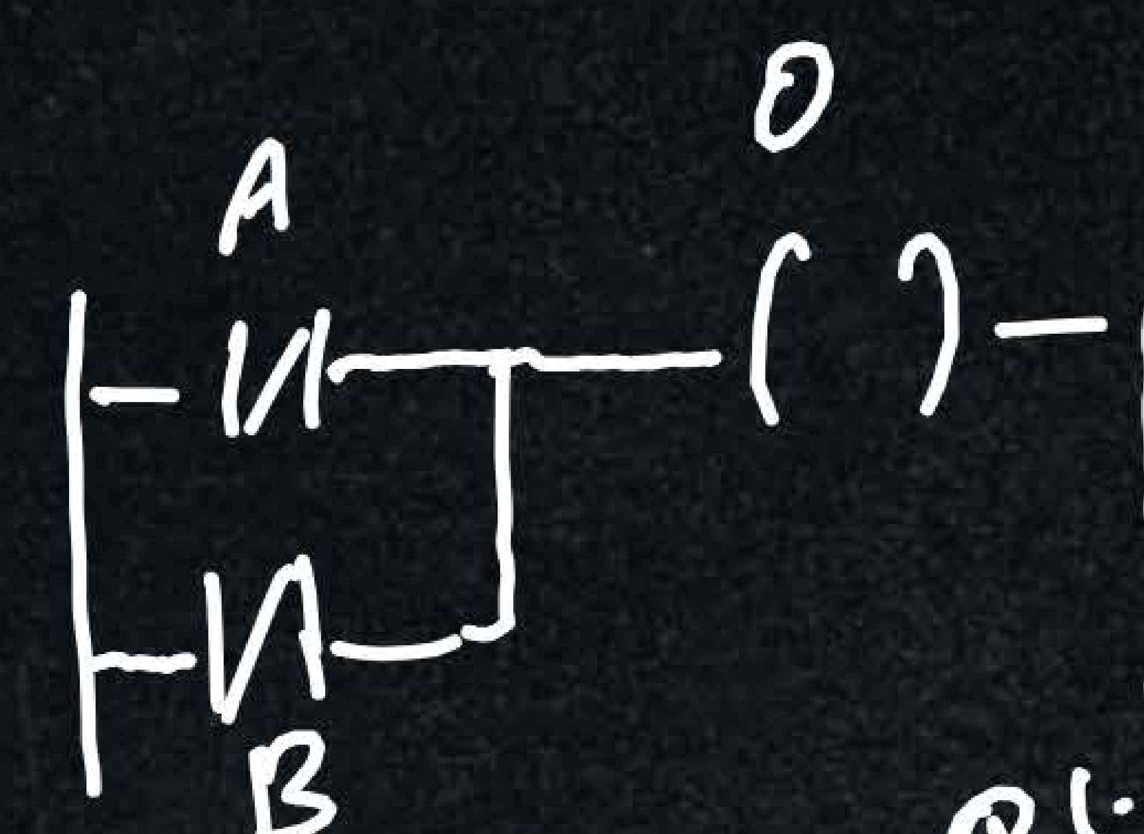


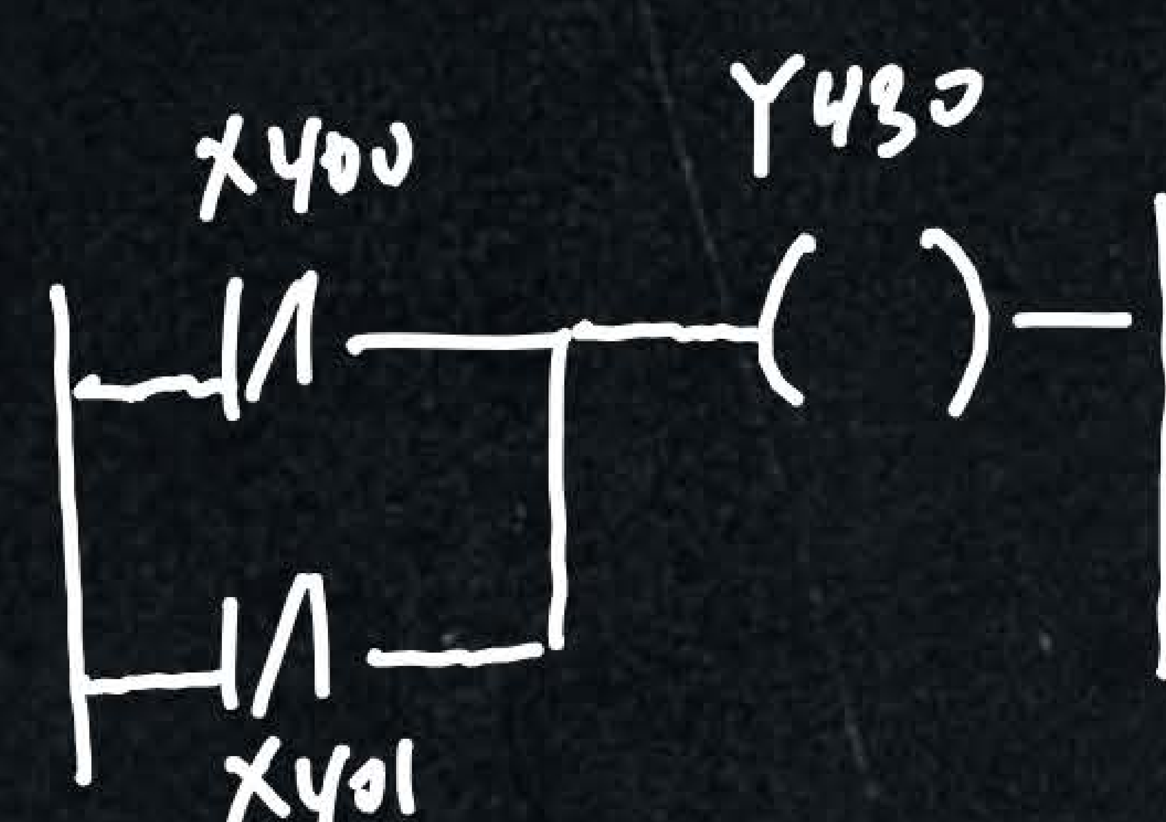


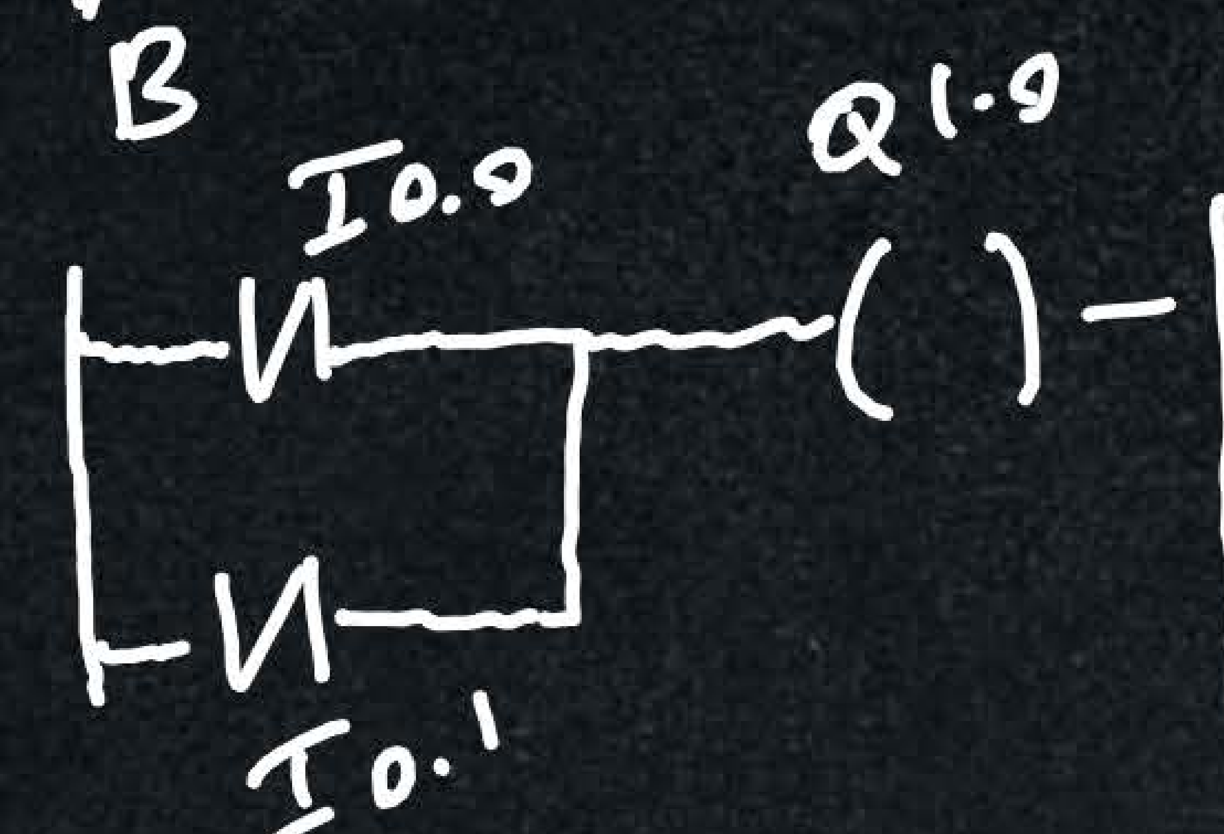
3) NAND

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

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



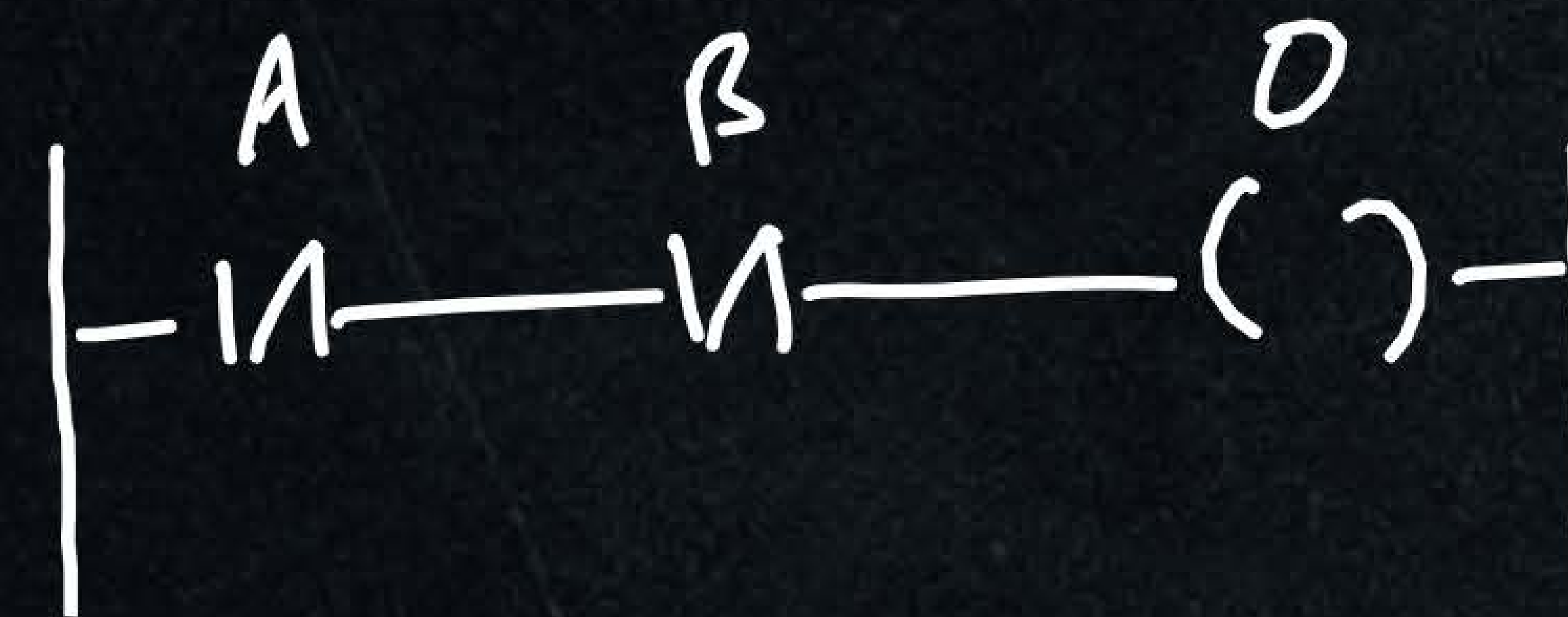


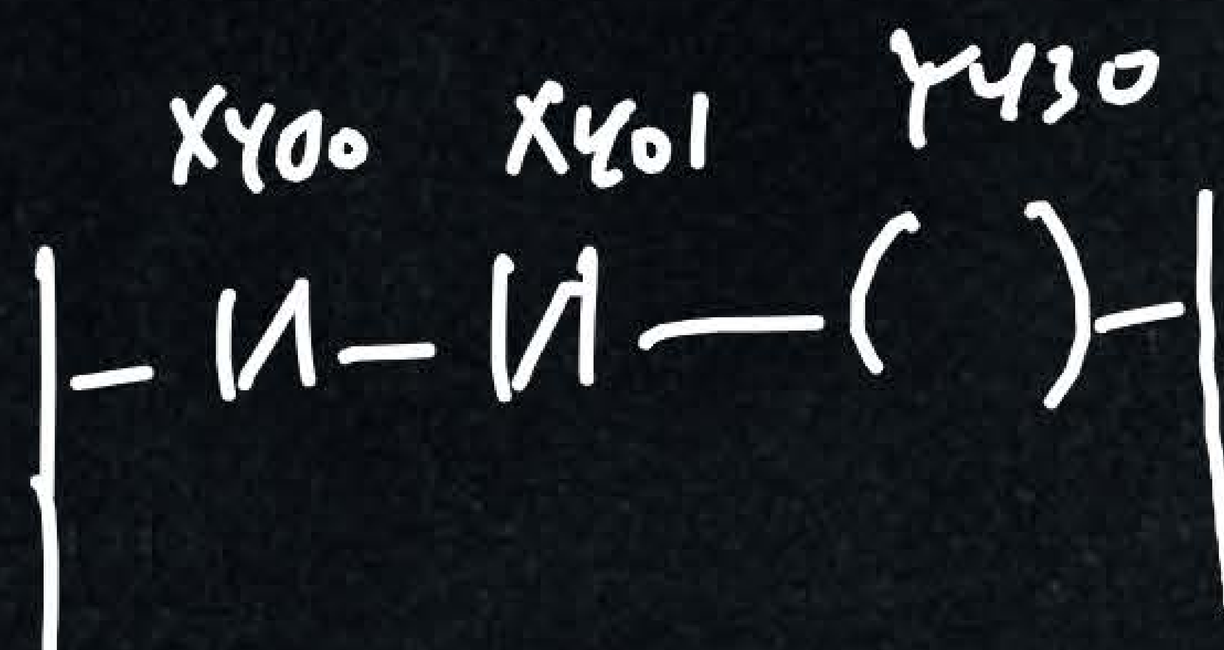


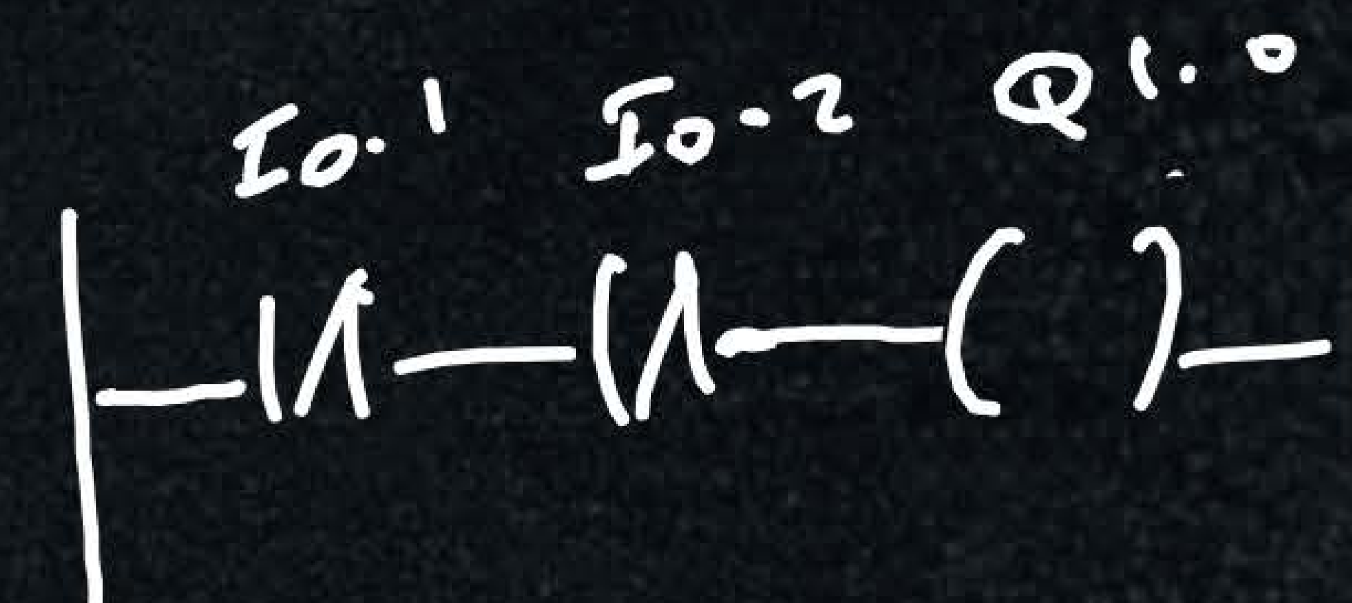
4) NOR

$O = (A+B)' = A'B'$

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



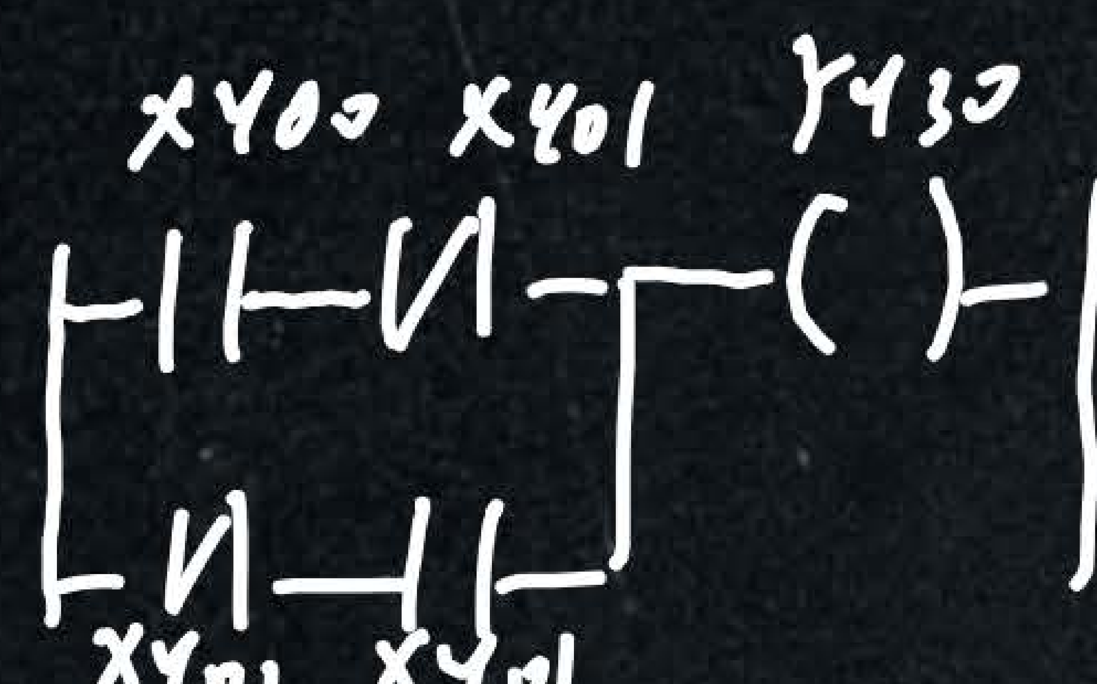


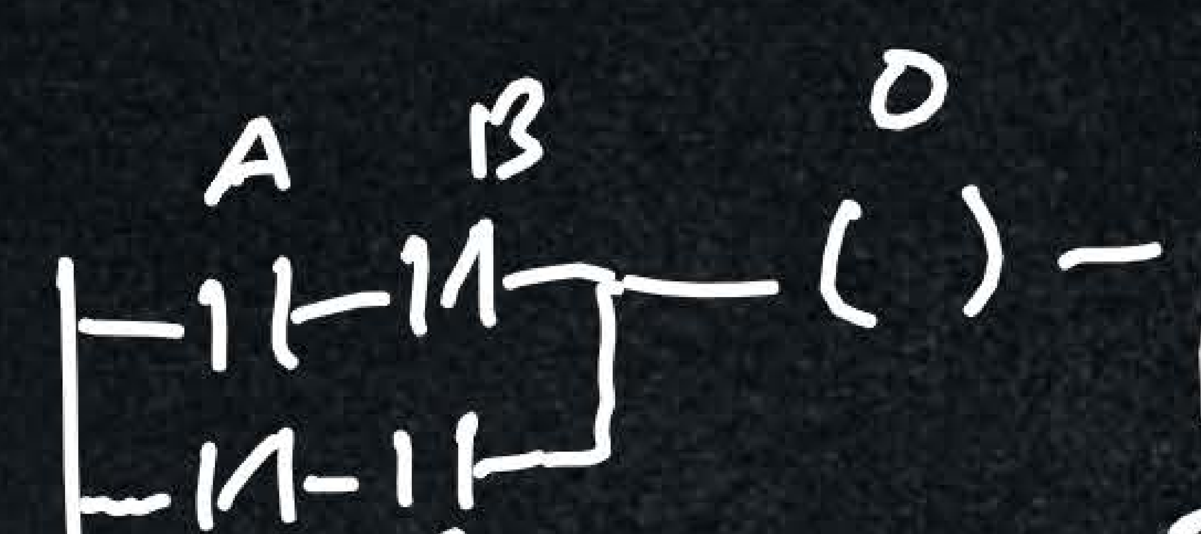


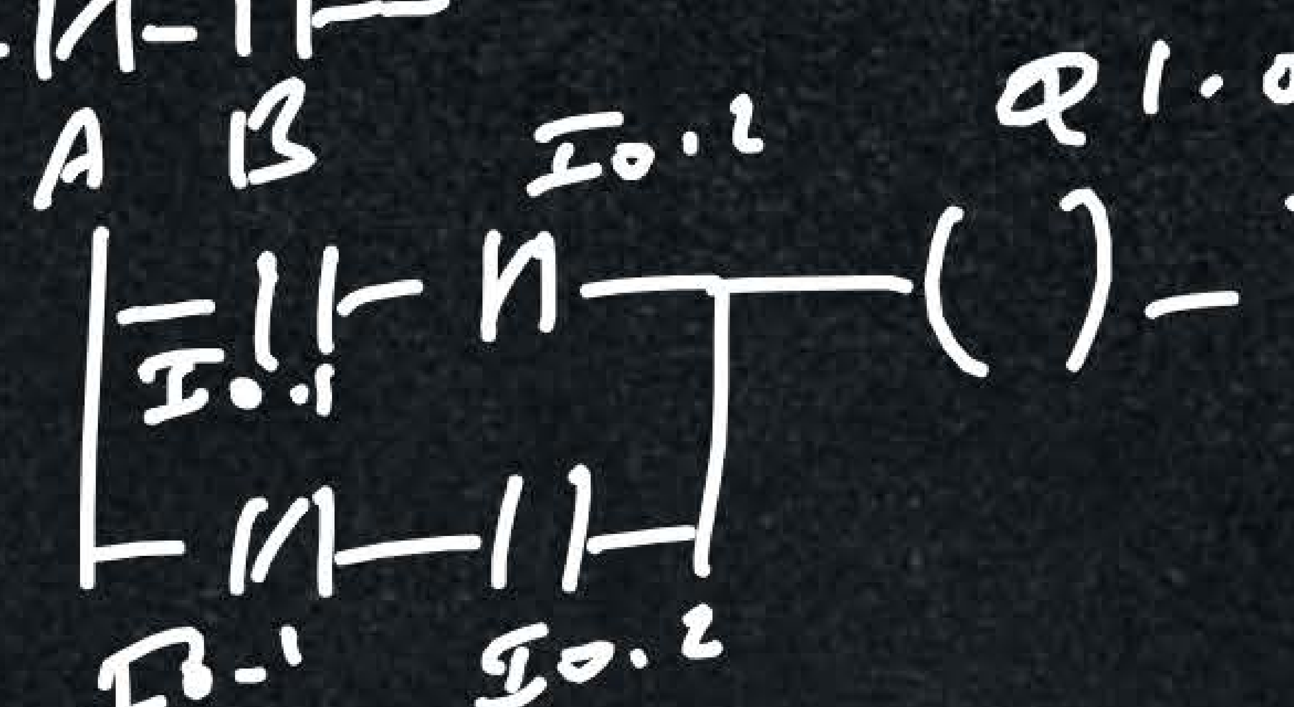
5) XOR

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

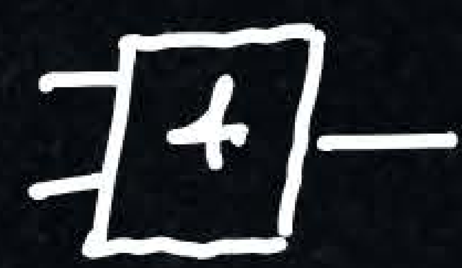
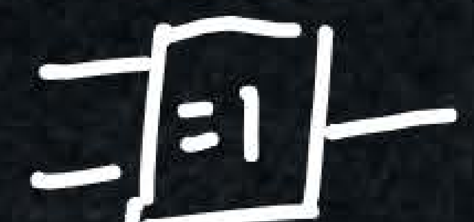
$O = AB' + A'B$







Logic Functions (FBD)

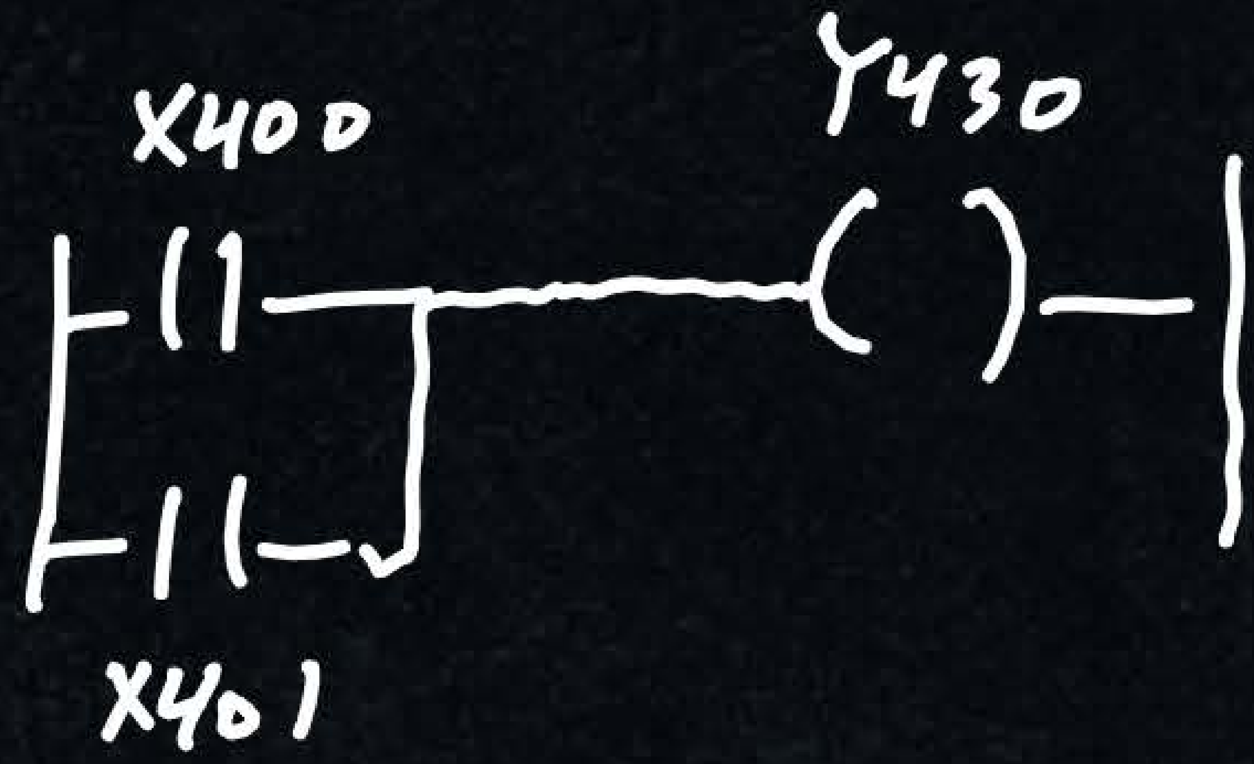
Function	FBD
AND	
OR	
NOT	
NAND	
NOR	
XOR	

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 element with open contacts
ORI	ON	" " " closed "
OUT	=	An output

Logic Functions (IL)

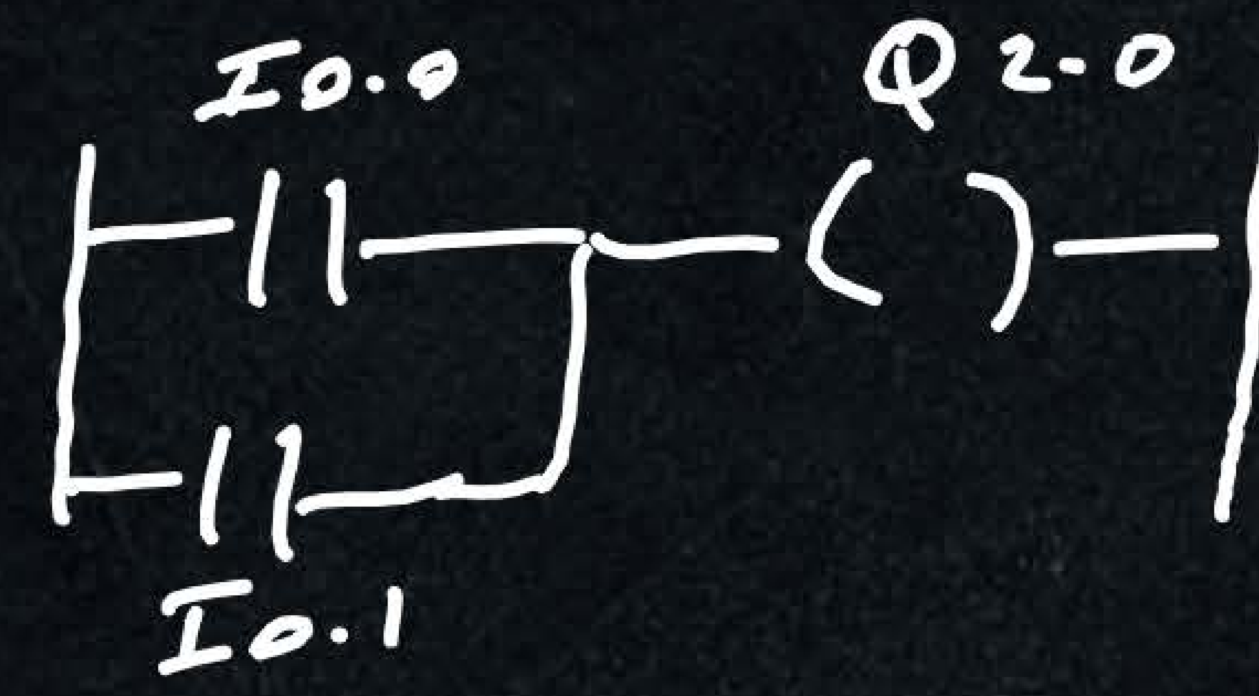
1) OR



LD X 400

OR X 401

OUT Y 430

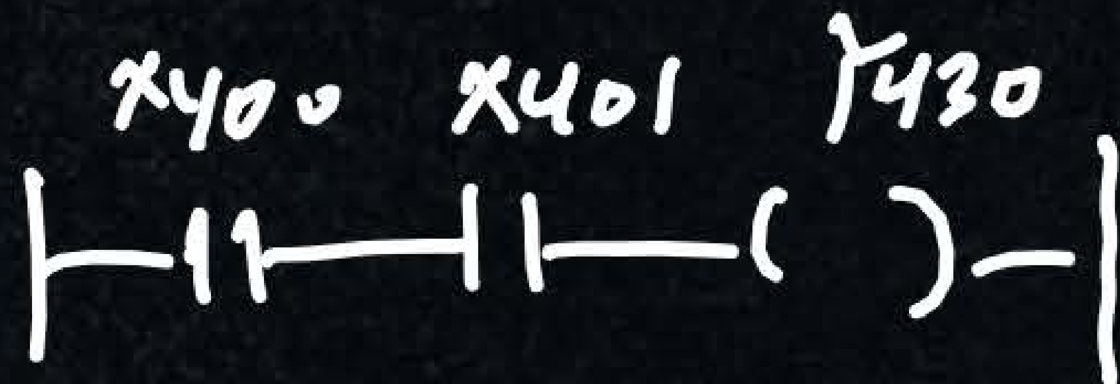


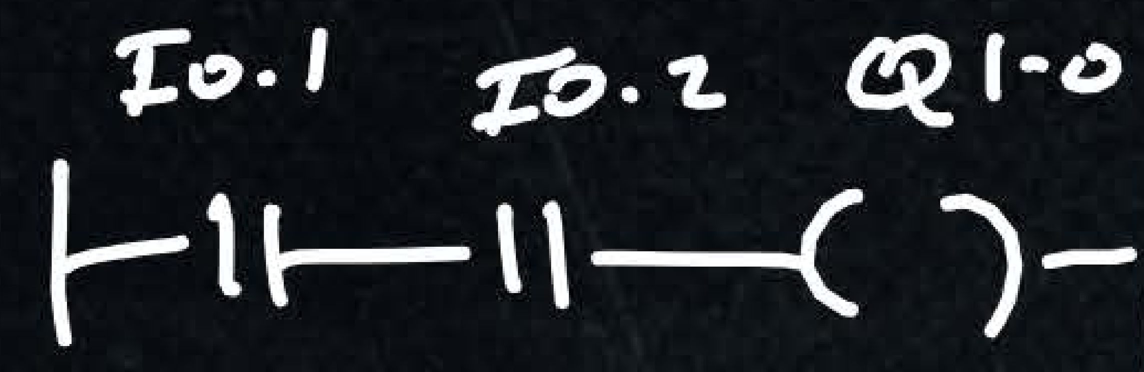
A I 0.0

O I 0.2

= Q 2.0

2) AND





LD X400

AND X401

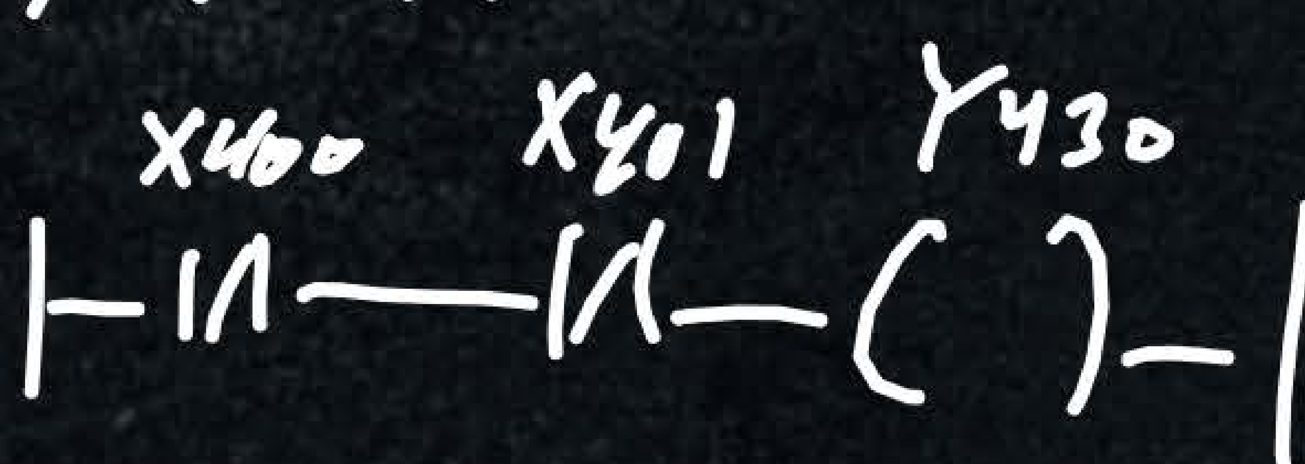
OUT Y430

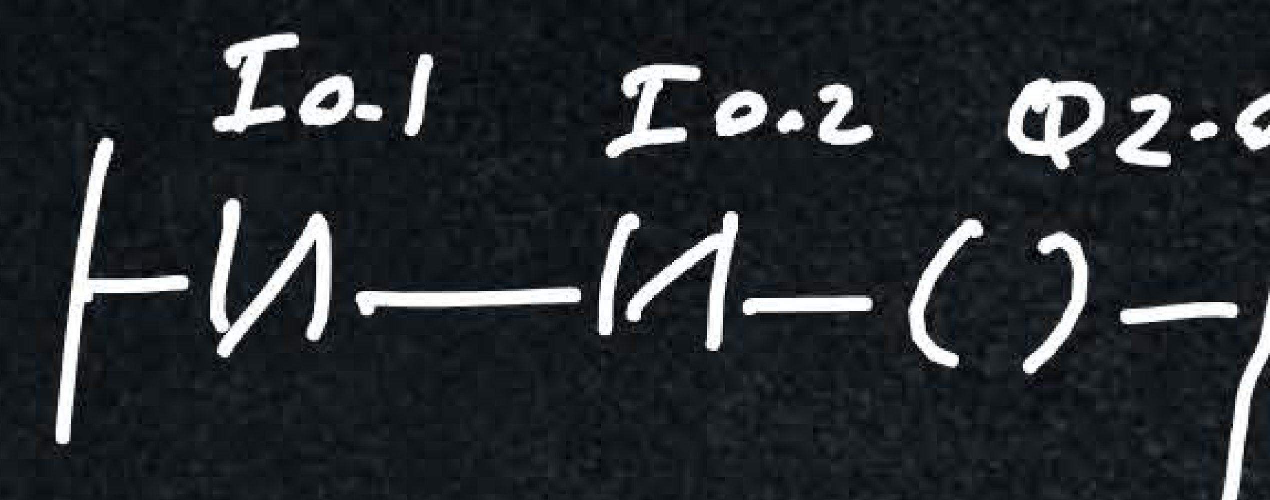
A I0.1

A I 0.2

= Q 1-0

3) NOR





LDI X400

ANI X401

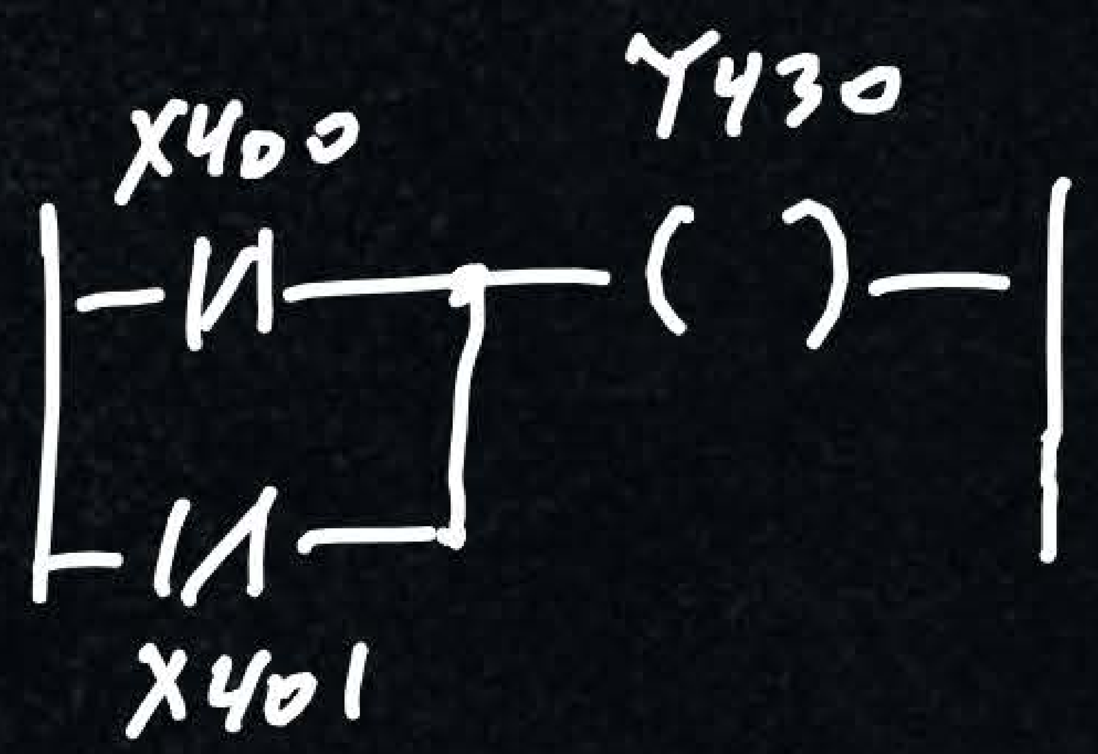
OUT Y430.

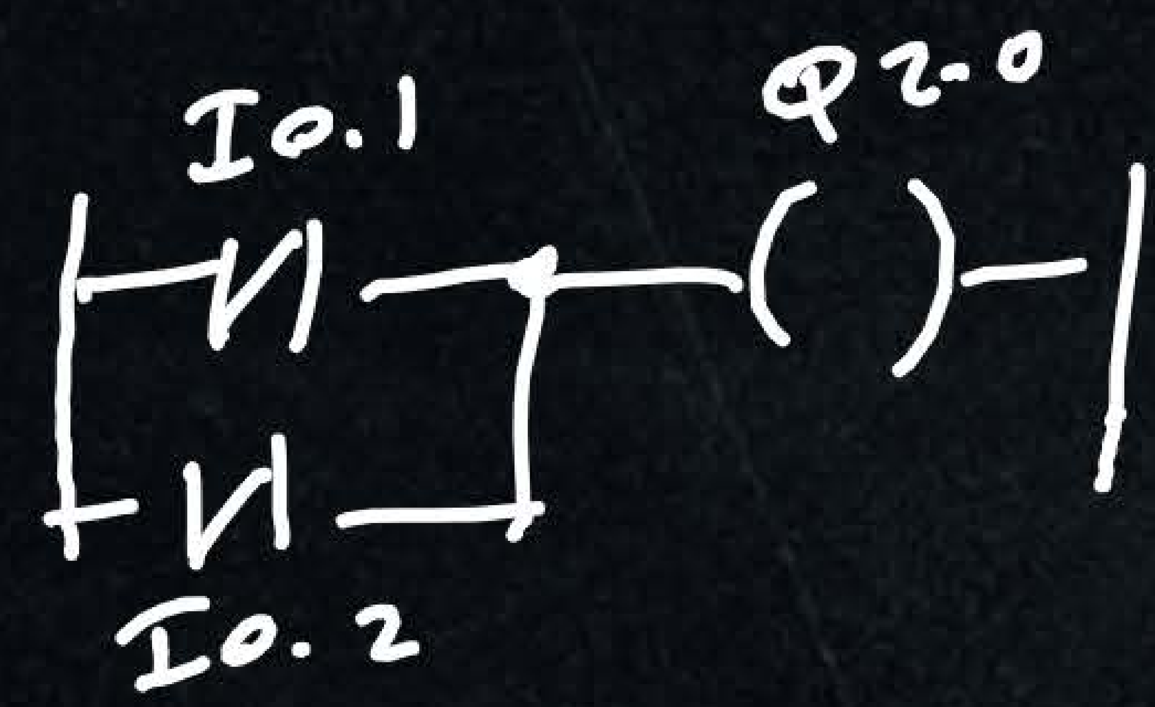
AN I0.1

AN I0.2

= Q 2-0

4) NAND





LDI X400

ORI X401

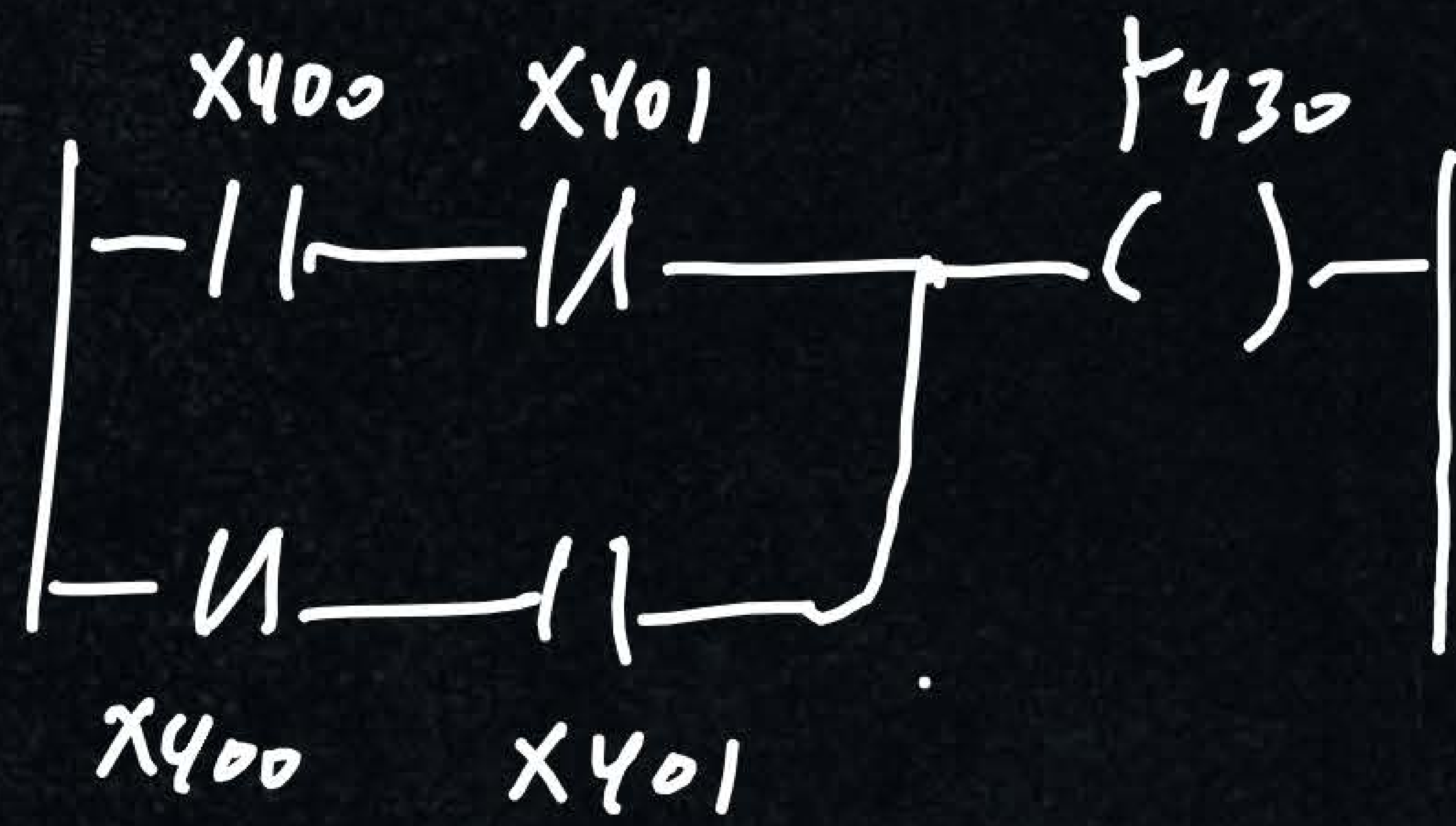
OUT Y430

AN I0.1

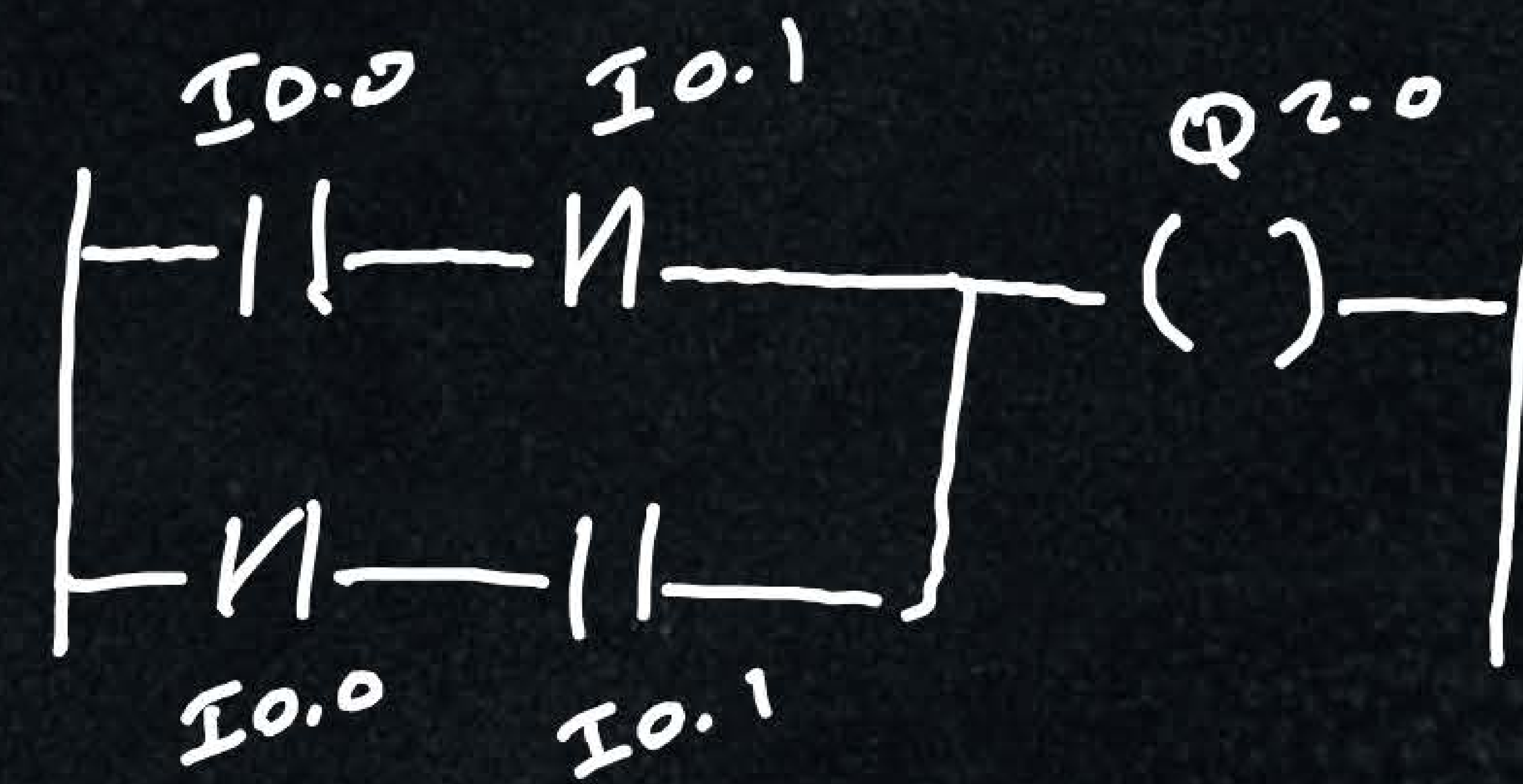
ON I0.2

= Q2.0

5) XOR




```
LD X400
ANI X401
LDI X400
AND X401
ORB
OUT Y430
```




```
A (
  A I0.0
  AN I0.1
)
O(
  AN I0.0
  A I0.1
)
=Q 2.0.
```

Example: Y-Δ starter of 3Φ IM using RLC

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

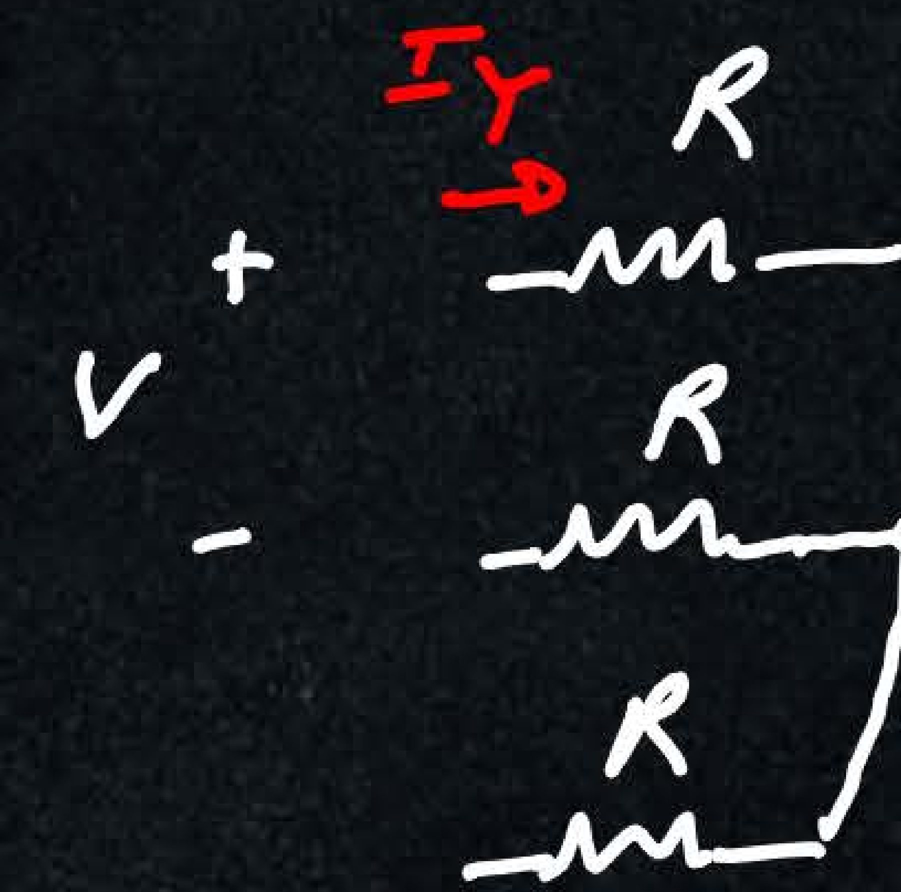
$$I_Y = \frac{V/\sqrt{3}}{R}$$

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

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

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

$$T_{el,Y} = \frac{1}{3} T_{el,\Delta}$$

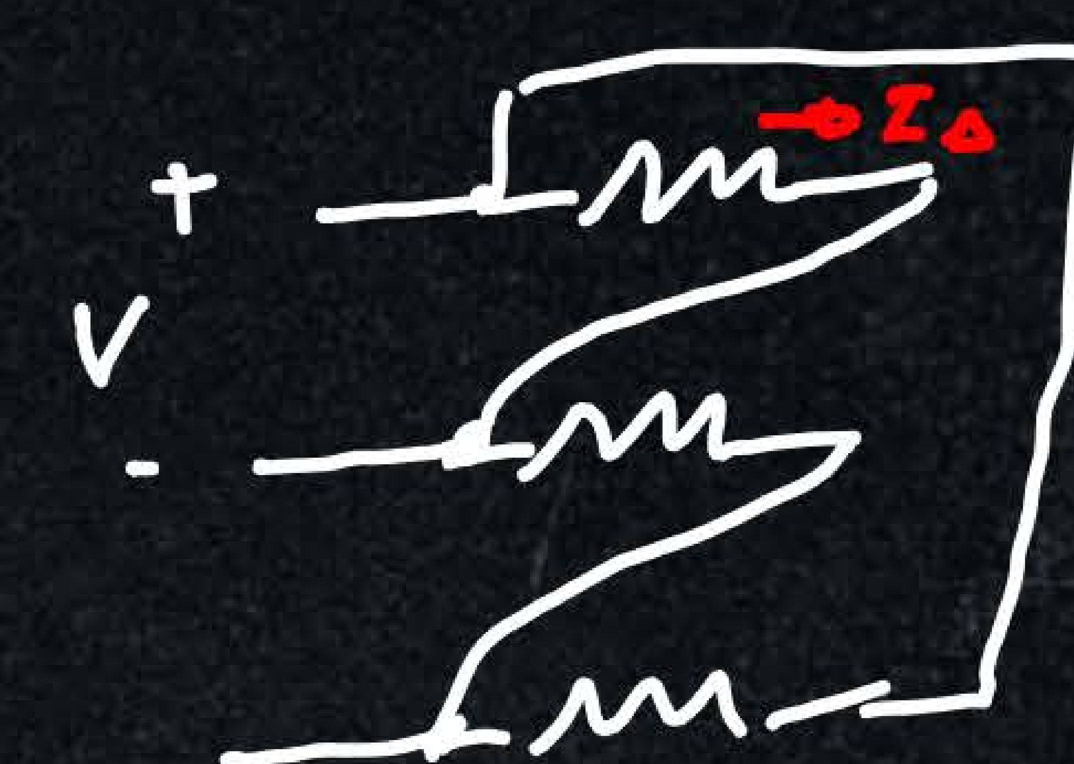


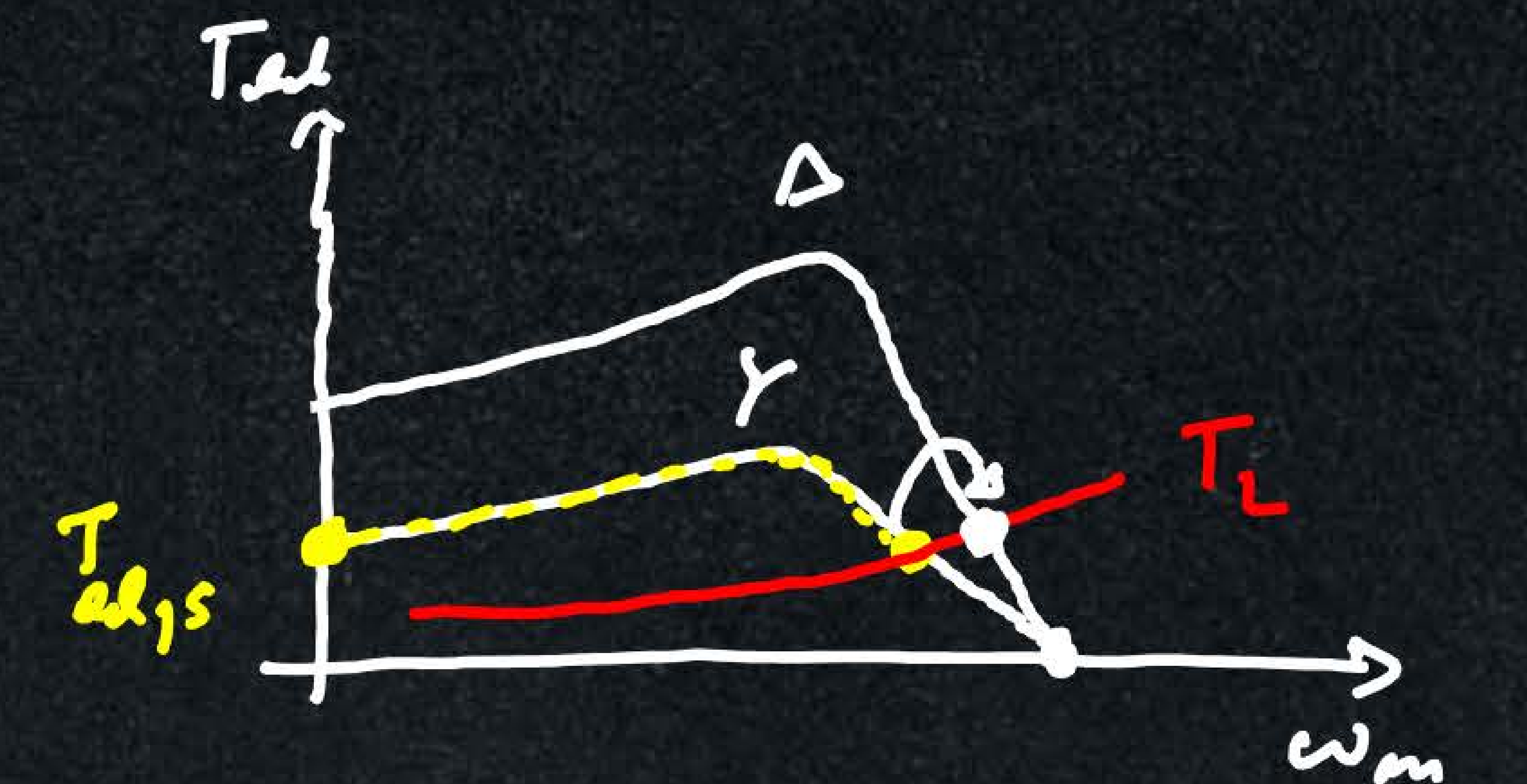
L-L RMS voltage supply

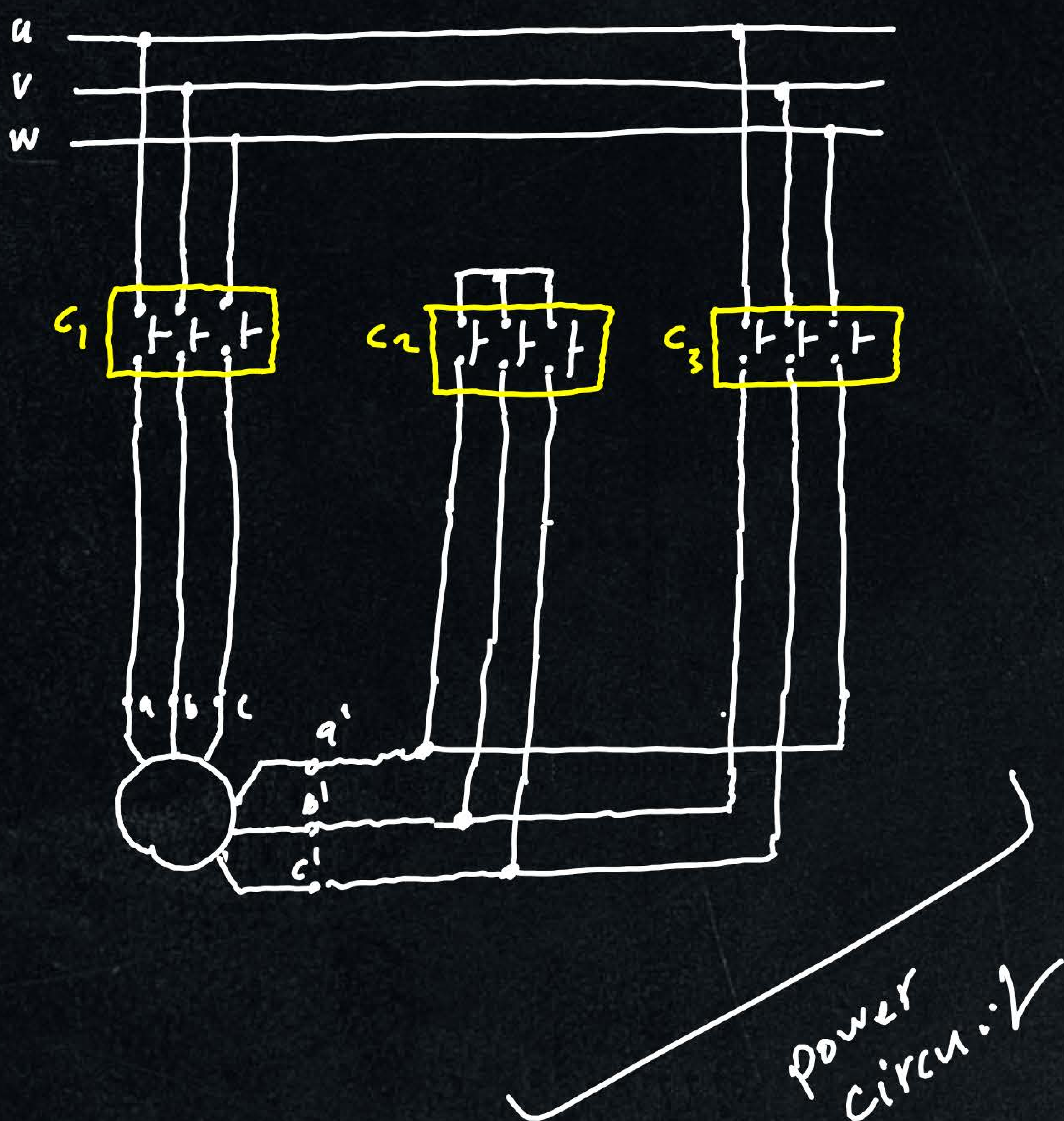
400 V

✓ 400/680 V (Δ / Y)

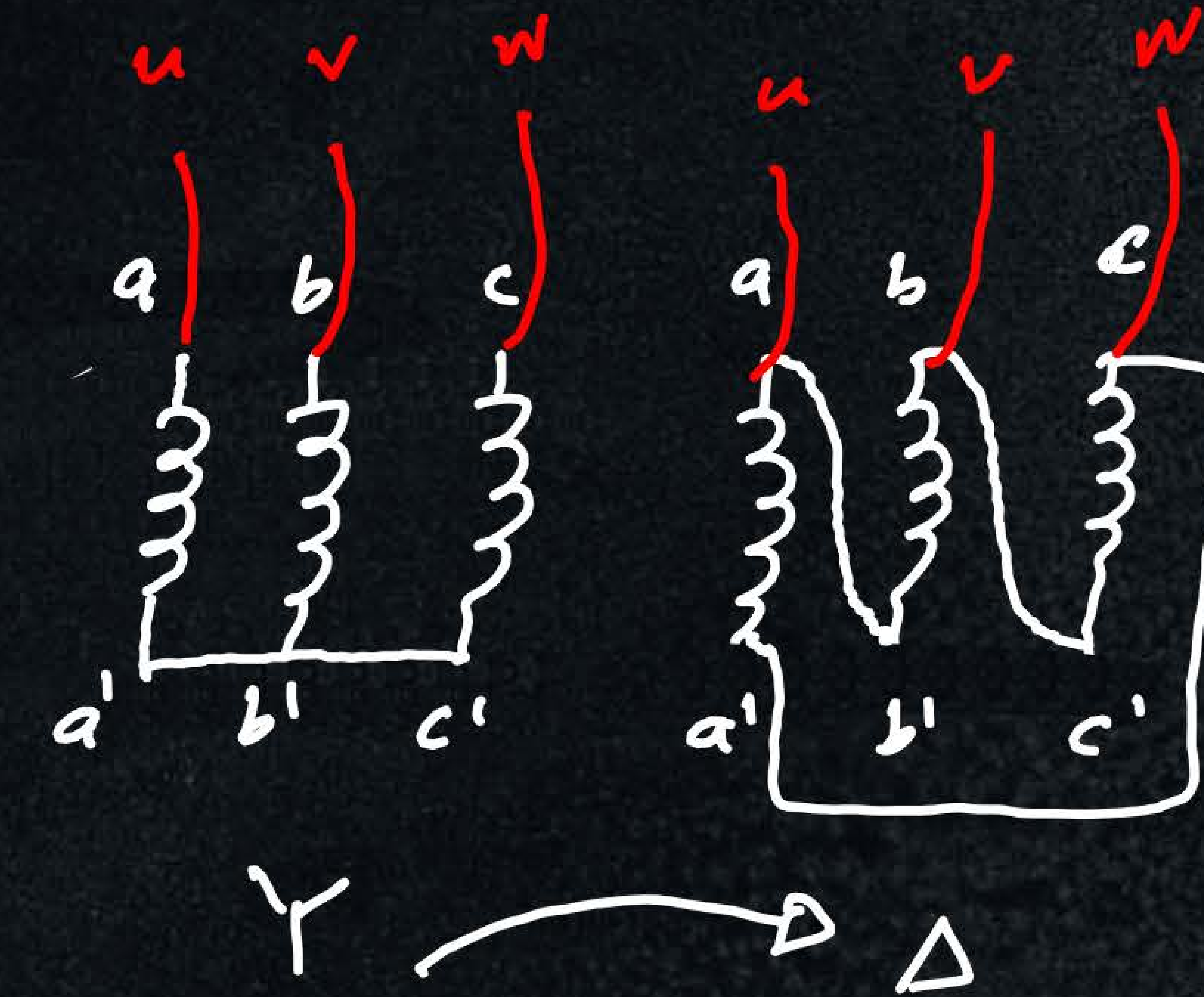
✗ 230/400 V (Δ / Y)





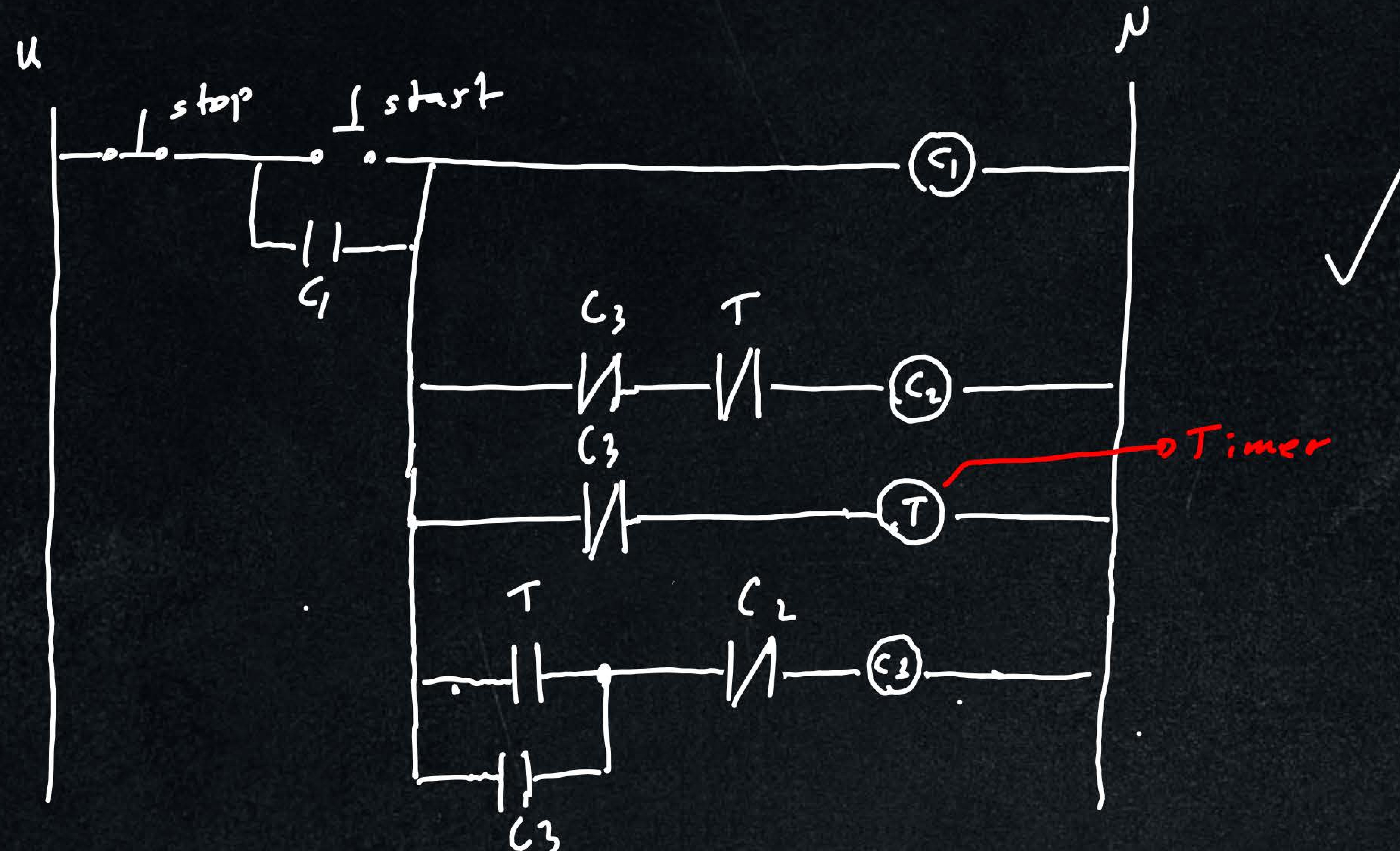


u
v
w
c1
c2
c3
a
b
c
a'
b'
c'
power
circuit

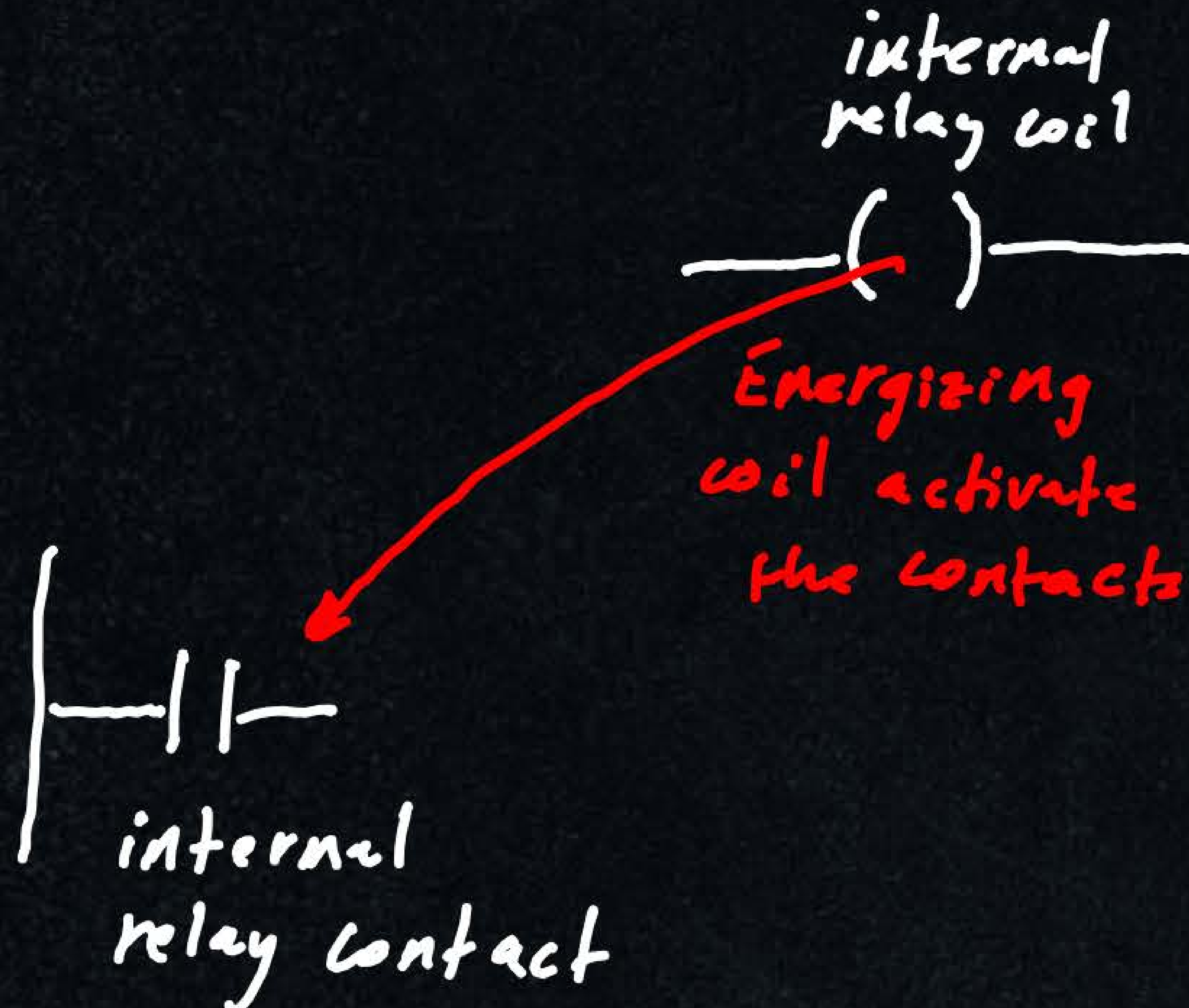


u v w
a b c
a' b' c'
u v w
a b c
a' b' c'
Y
Δ

Control Circuit



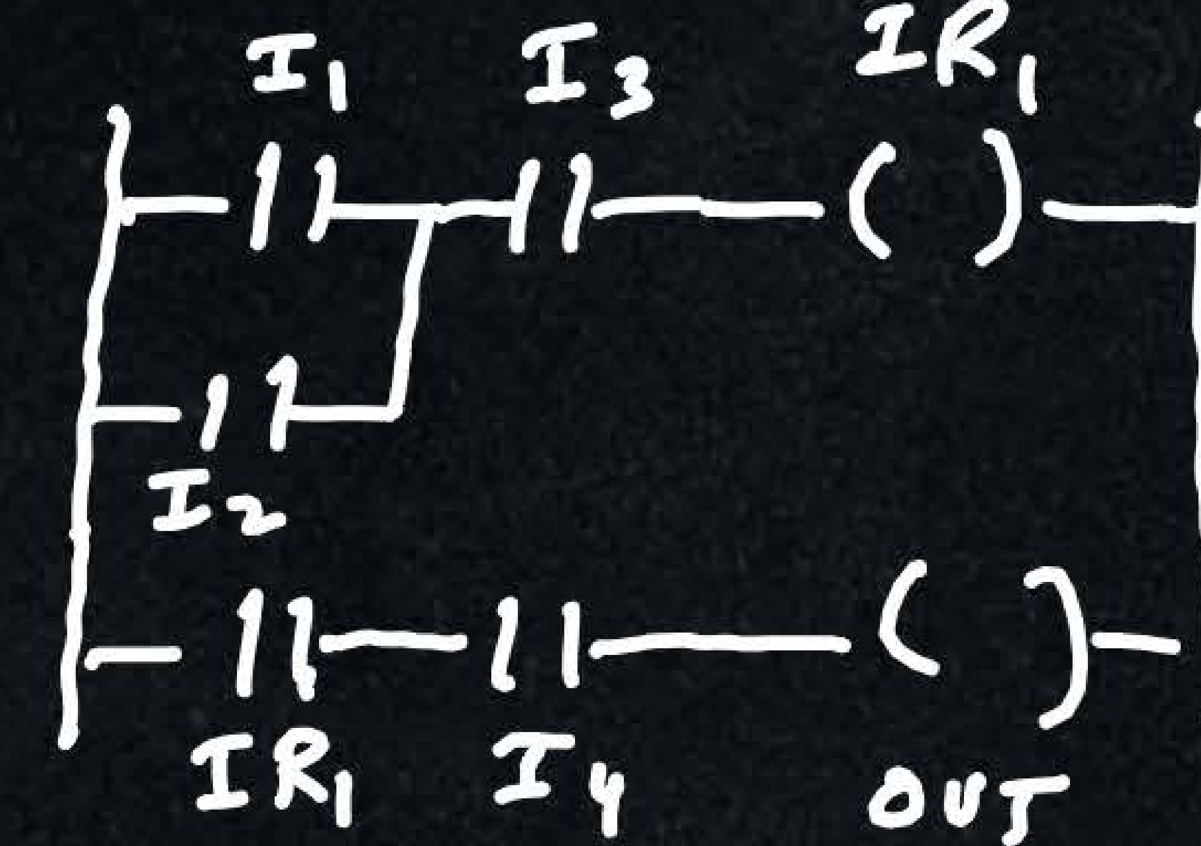
Internal Relay



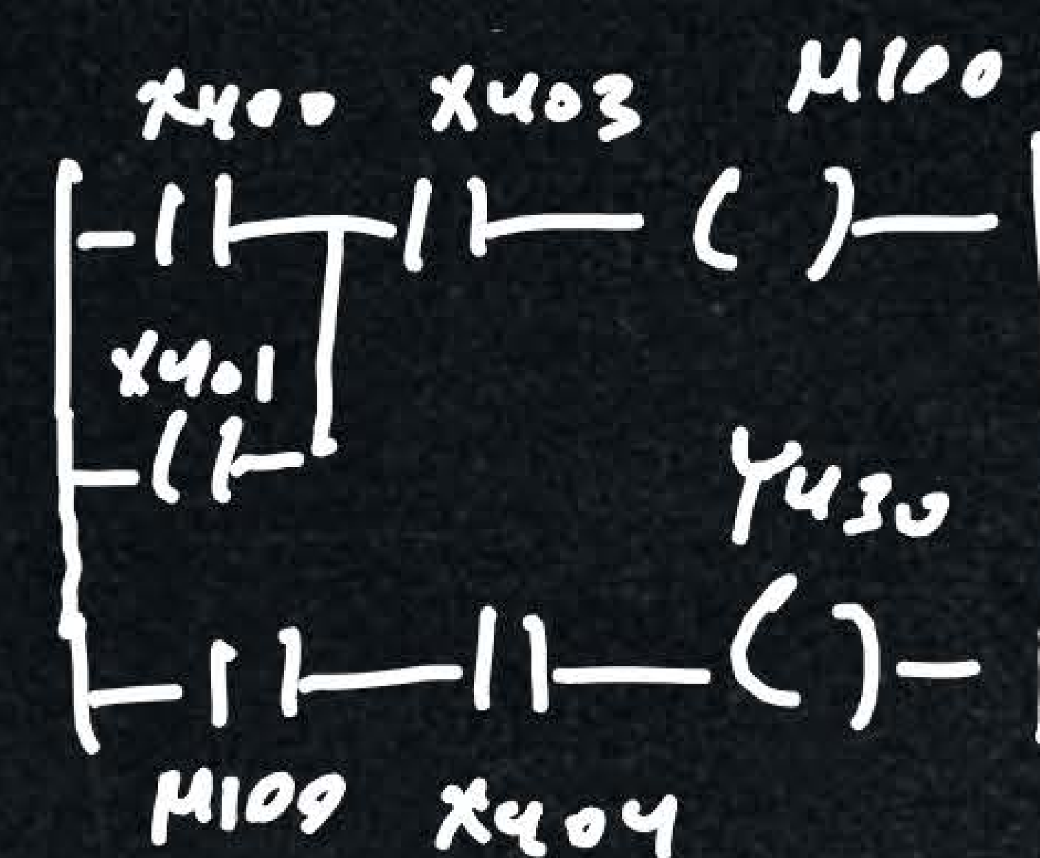
Addresses:-

Mitsubishi	M100, M101, ...
Siemens	F0.0, F0.1, ...
Toshiba	R000, R001, ...

Ex:



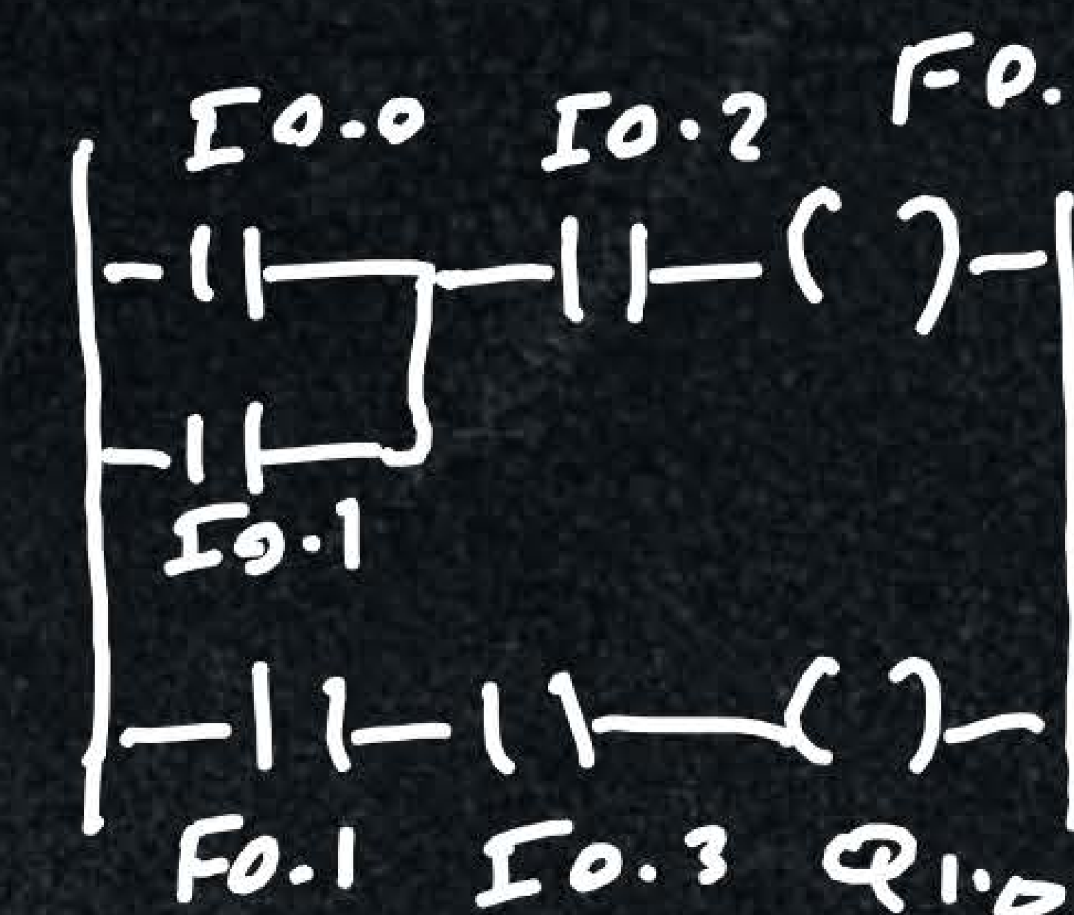
Mitsubishi:




```
LD X400
OR X401
AND X403
OUT M100

LD M100
AND X404
OUT Y430
```

Siemens

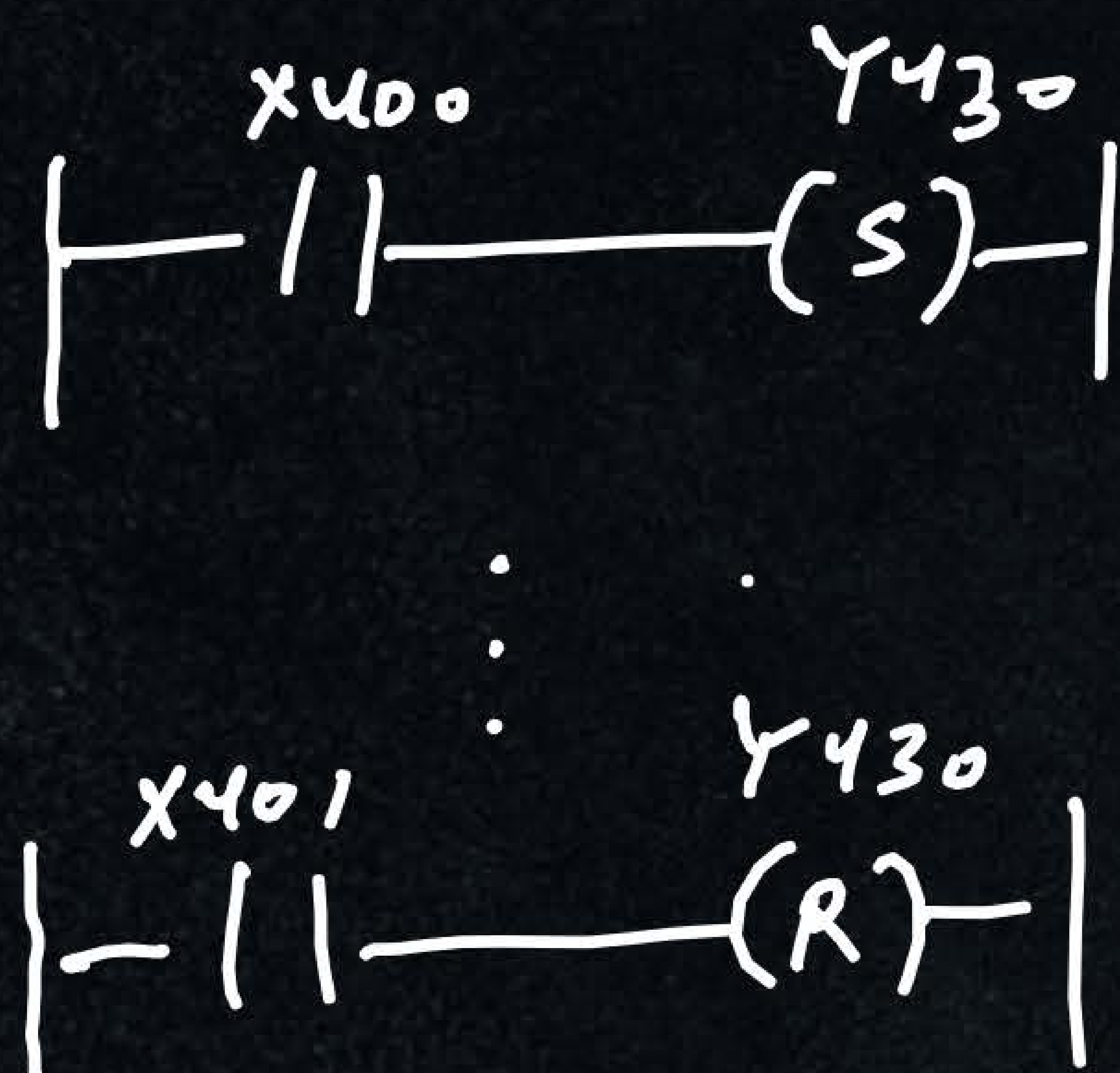


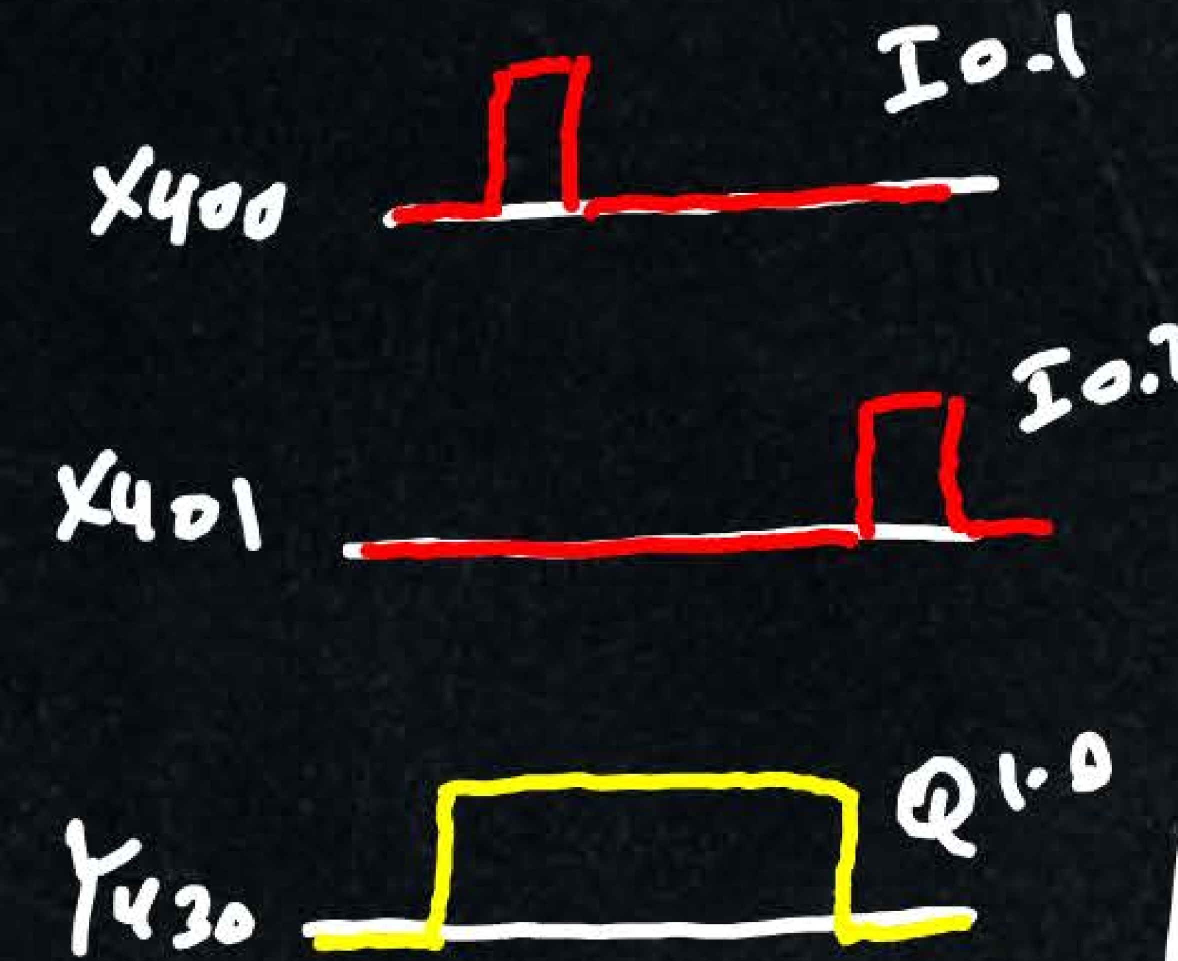

```
A I0.0
O I0.1
A I0.2
= F0.1

A F0.1
A I0.3
= Q1.0
```

Set and Reset

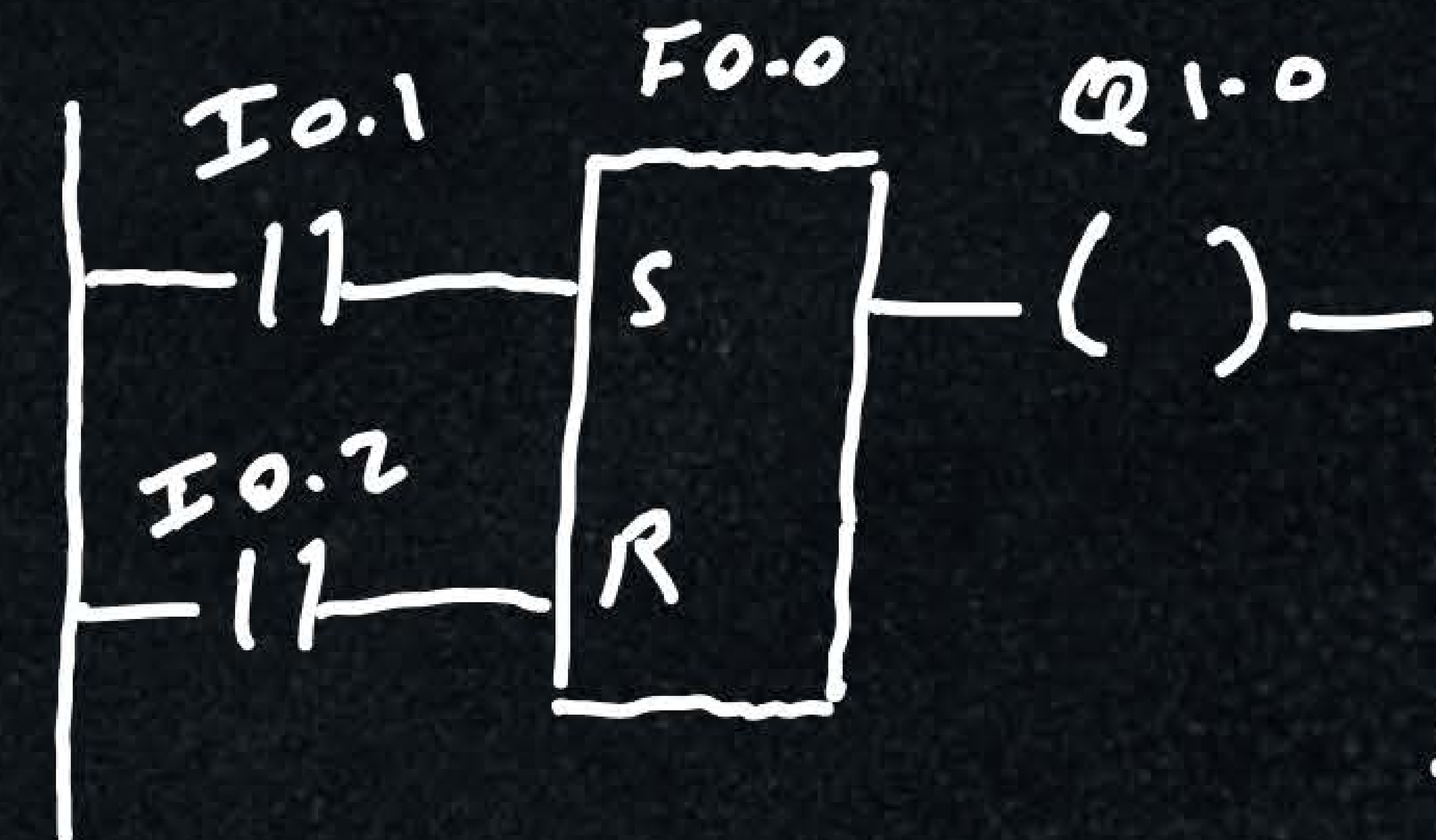
Mitsubishi:






```
LD X 400
S Y 430
⋮
LD X 401
S Y 430
```

Siemens

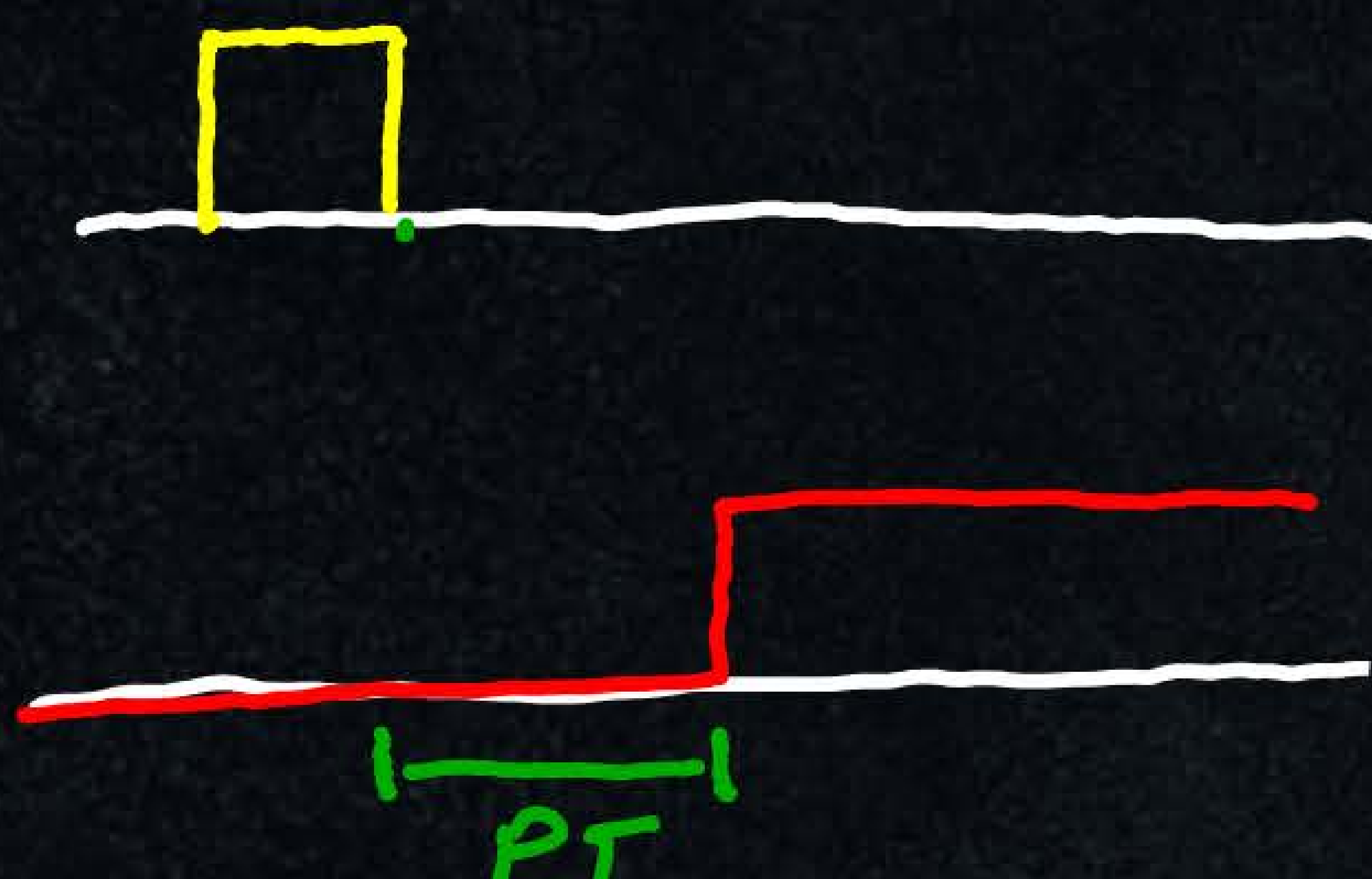



```
A I 0.1
S F 0.0
A I 0.2
R F 0.0
A F 0.0
= Q 1.0
```

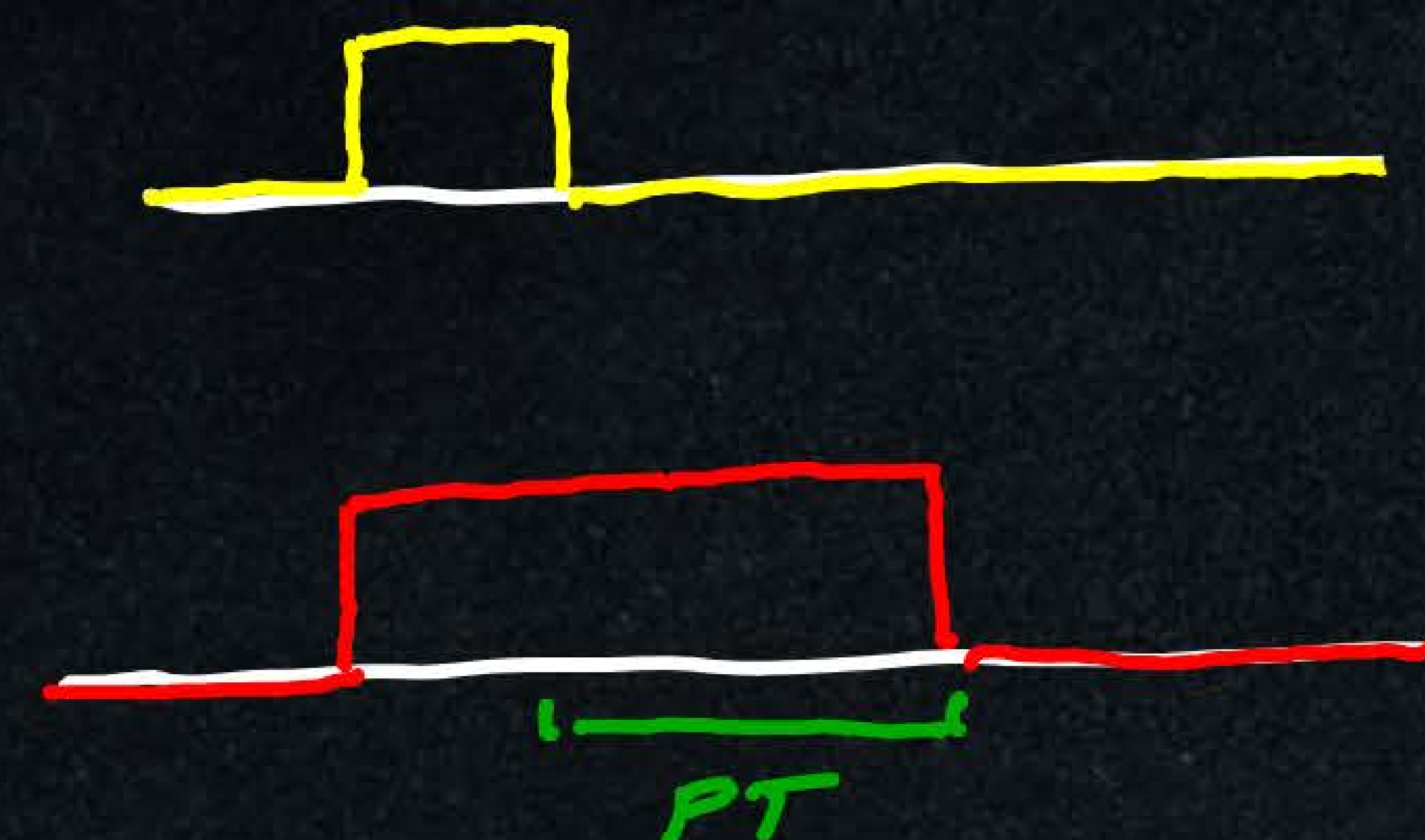
Timers

Types of timers

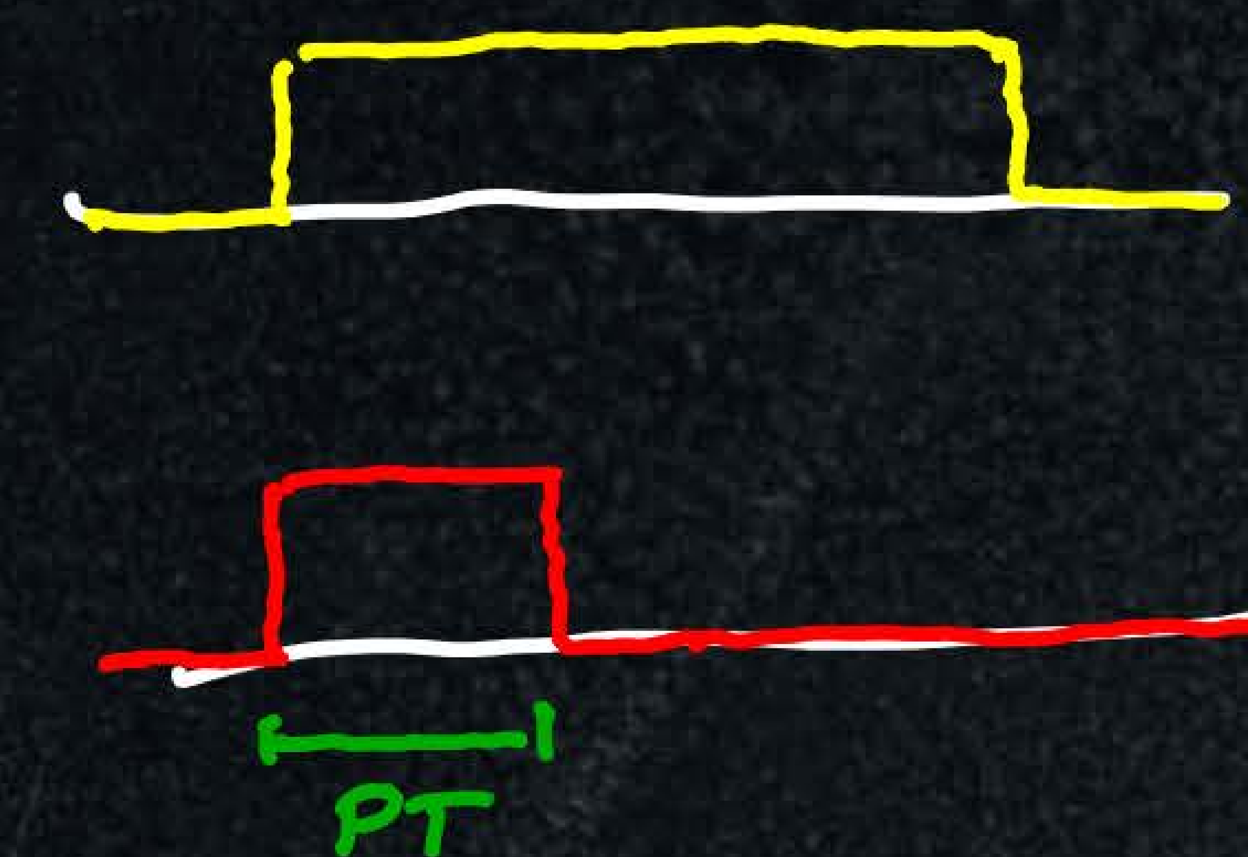
1) ON-delay timer



2) OFF-delay timer



3) pulse timer



PT: preset time

Mitsubishi

PT = n × time base → 10 ms, 100 ms, 1 sec, 10 s, and 100 s

where n is real integer

Siemens

KT XXX.X

000   0 → 10 ms
001   1 → 100 ms
002   2 → 1000 ms
⋮   ⋮

KT 5.2

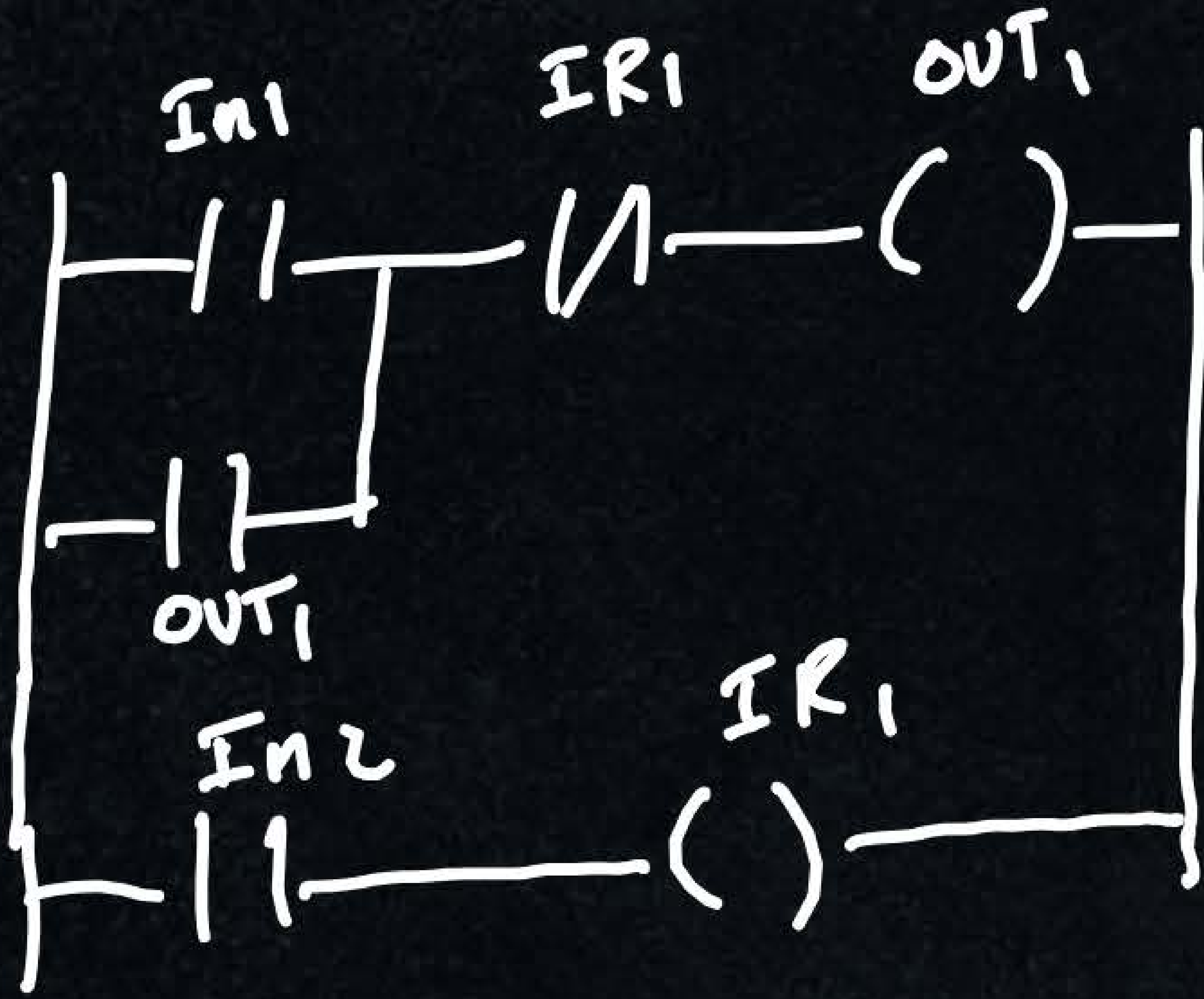
PT = 5 × 1000 ms
= 5 sec

Addresse

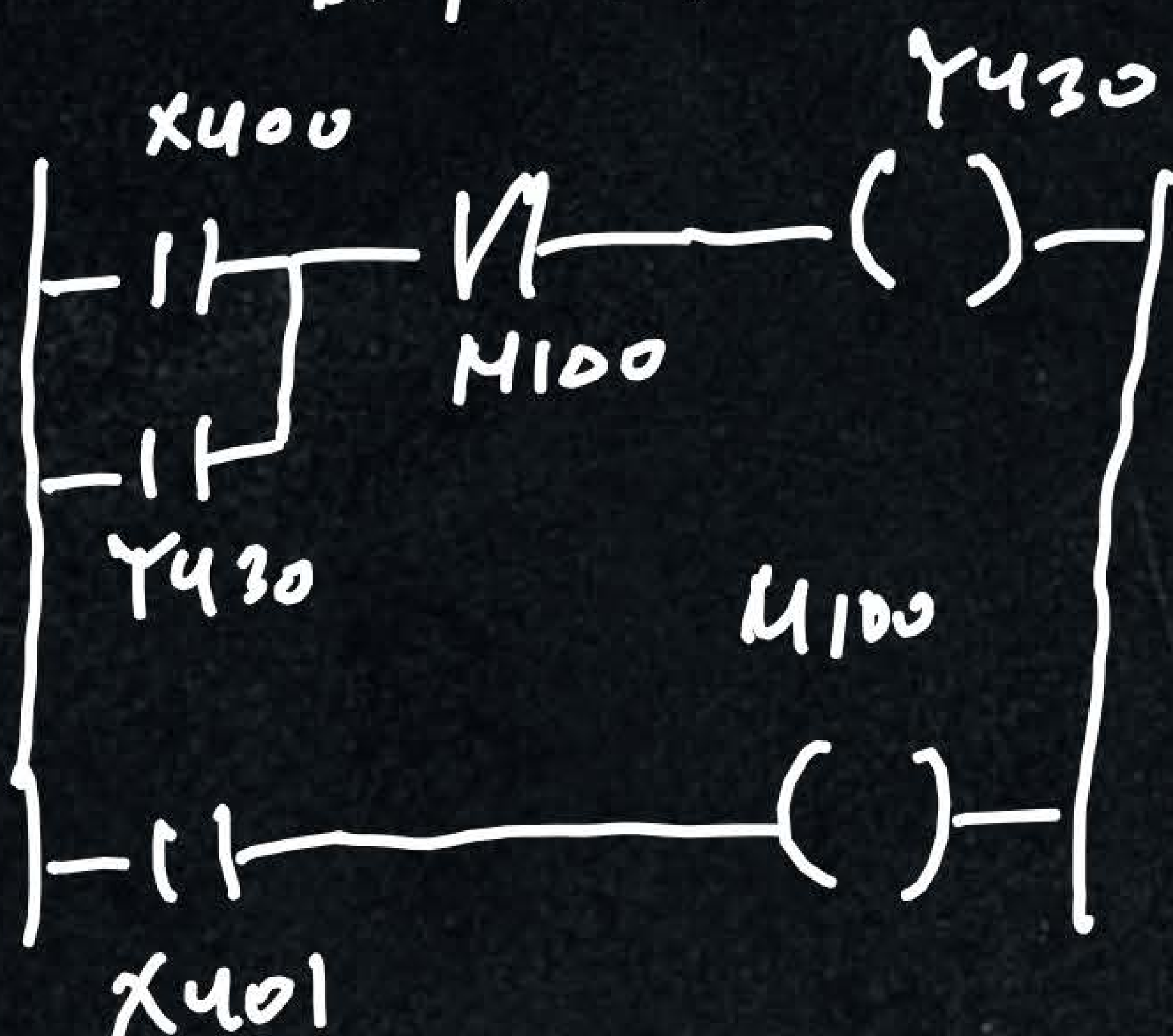
Mitsubishi : T450, T451, ...

Siemens : T0, T1, ...

Ex :- Resetting Latch

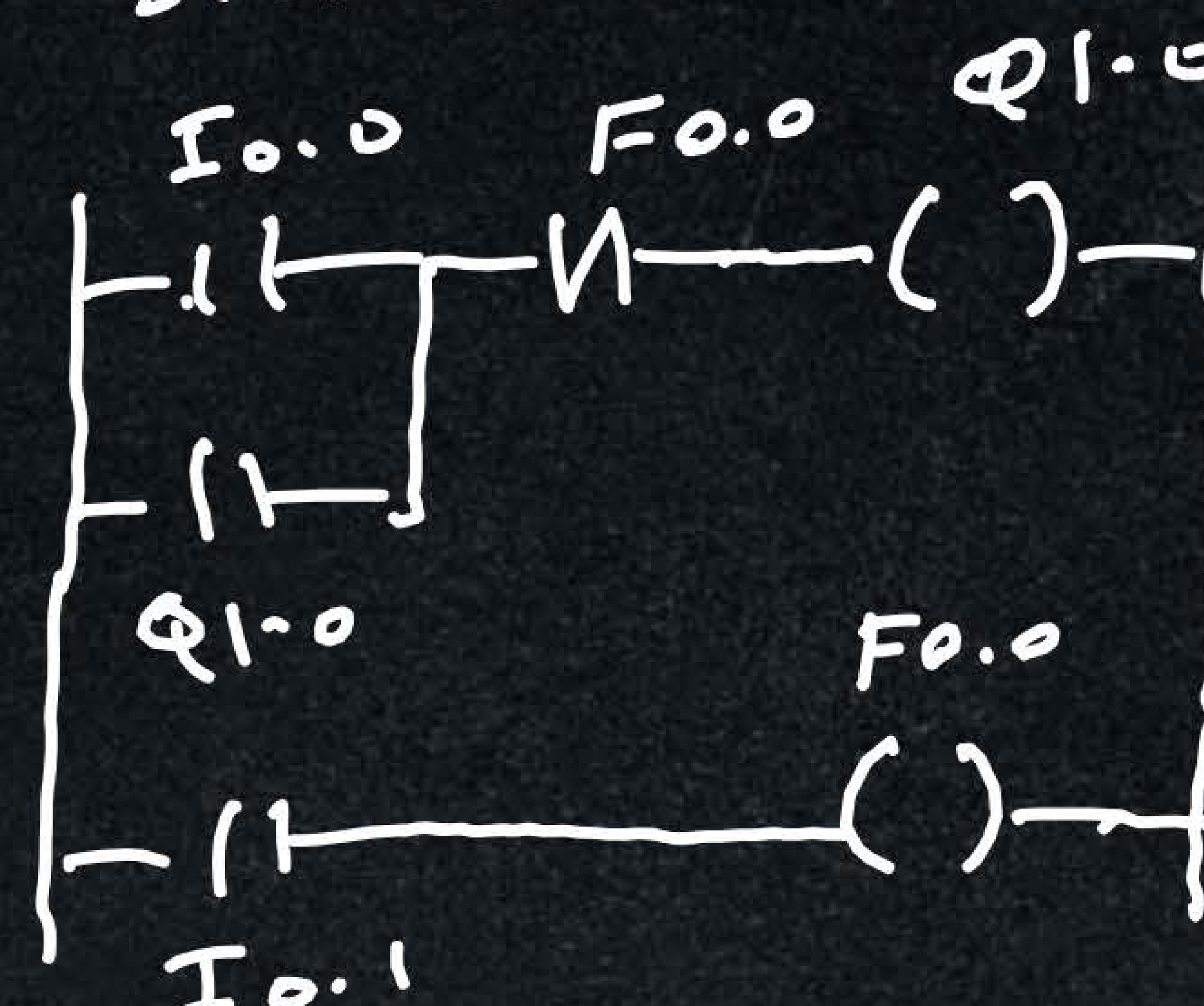


Mitsubishi



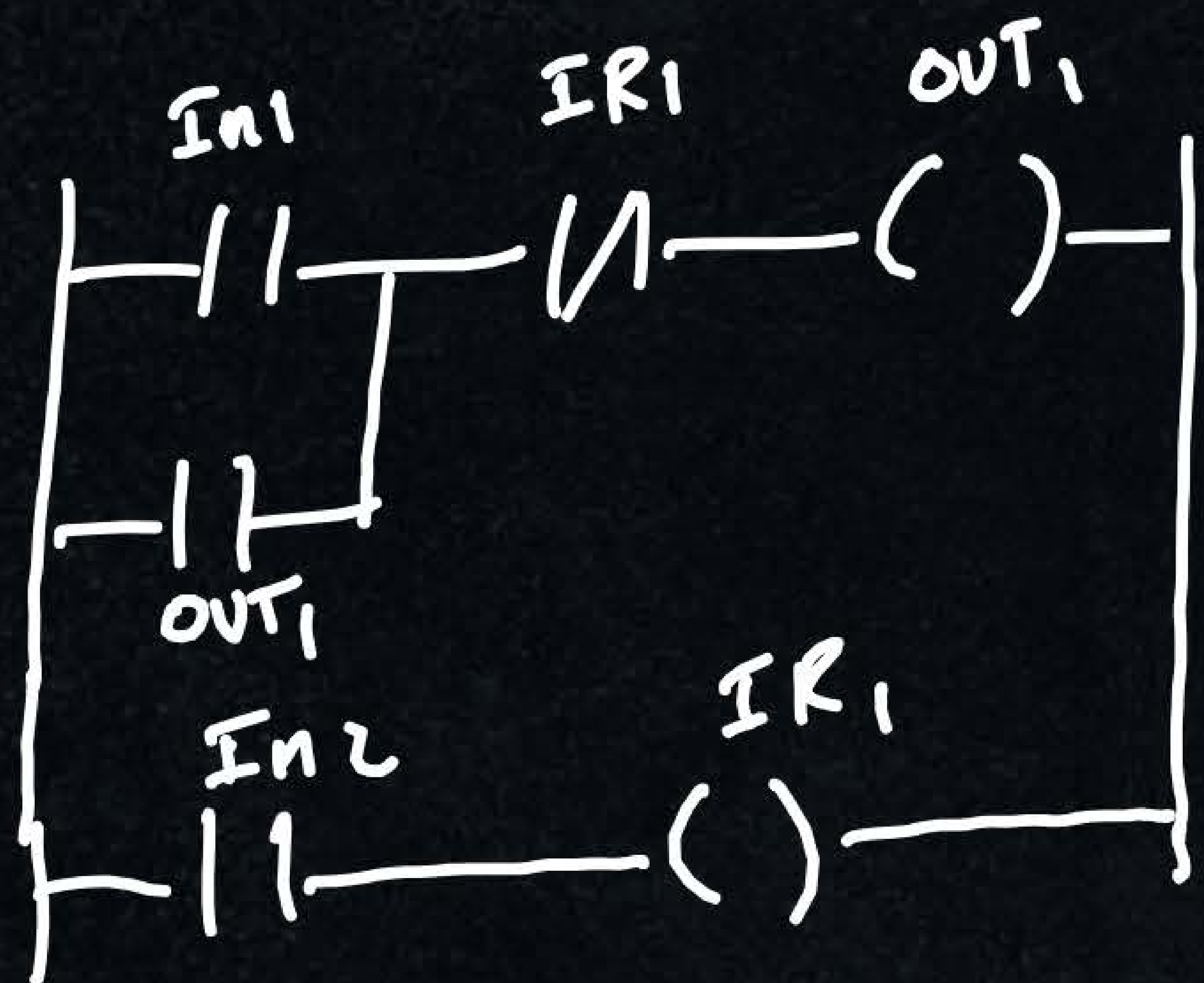
LD X400
OR Y430
ANI M100
OUT Y430
LD X401
OUT M100

Siemens

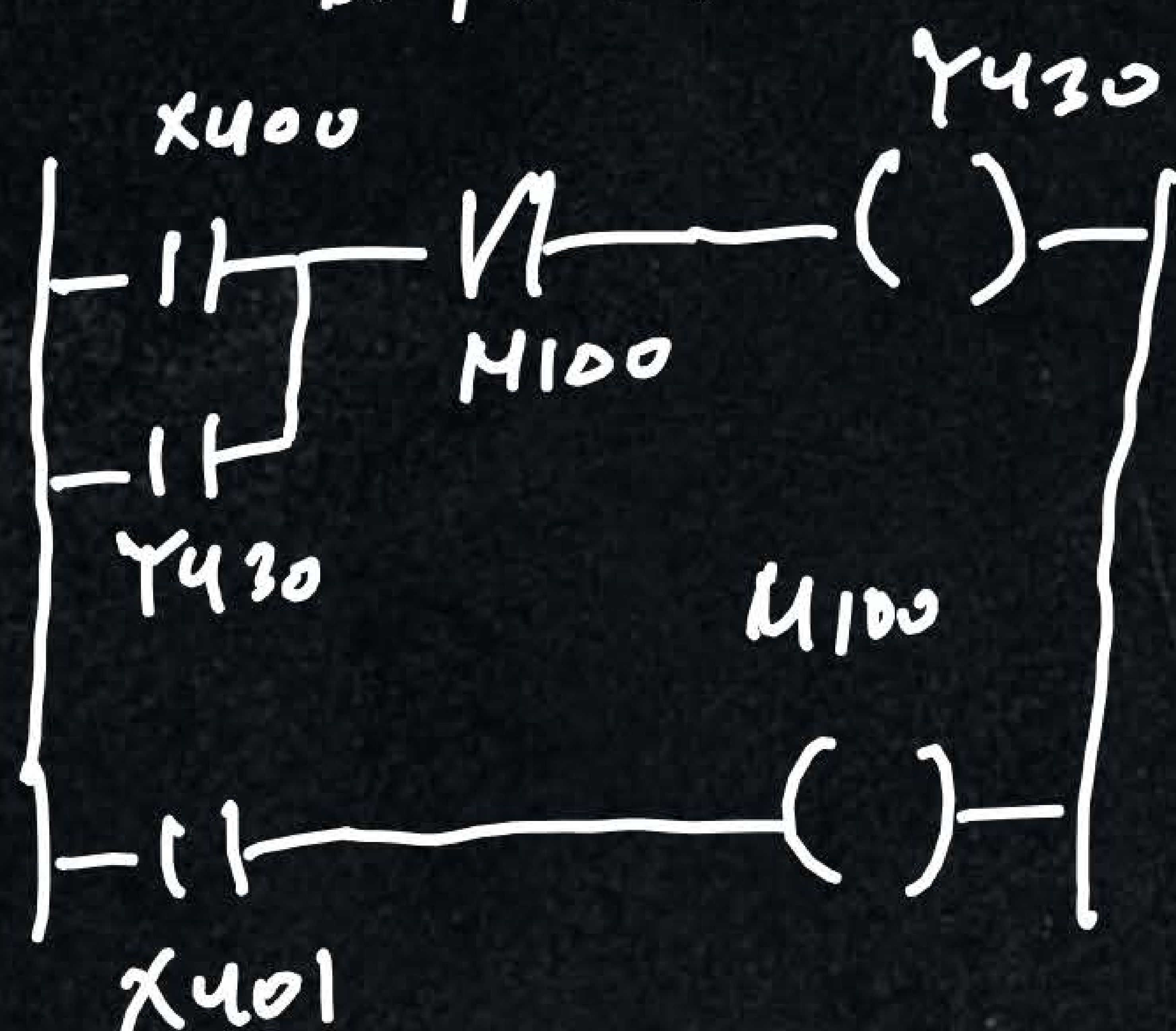


A I0.0
O Q1.0
AN F0.0
= Q1.0
A I0.1
= F0.0

Ex :- Resetting Latch

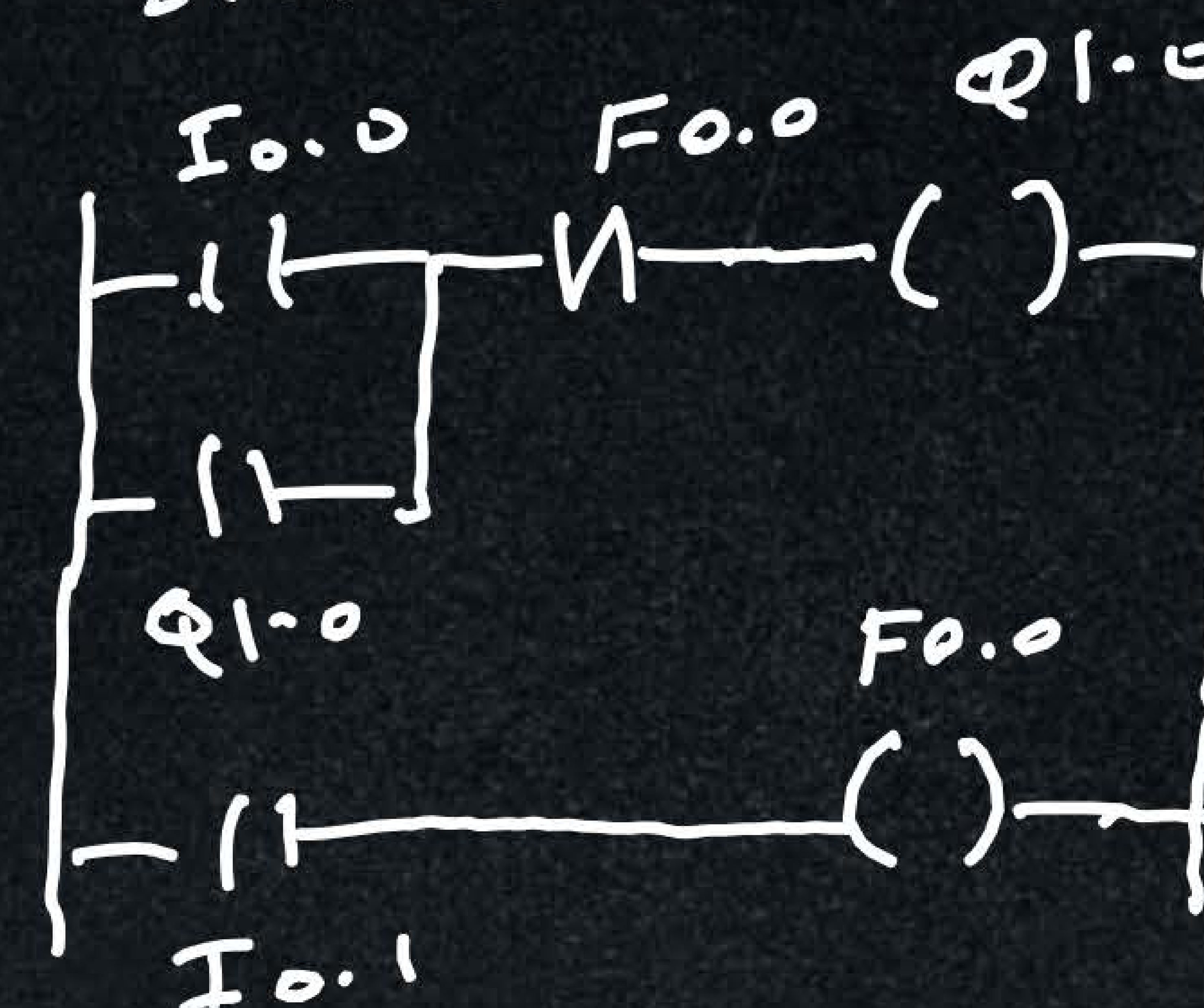


Mitsubishi



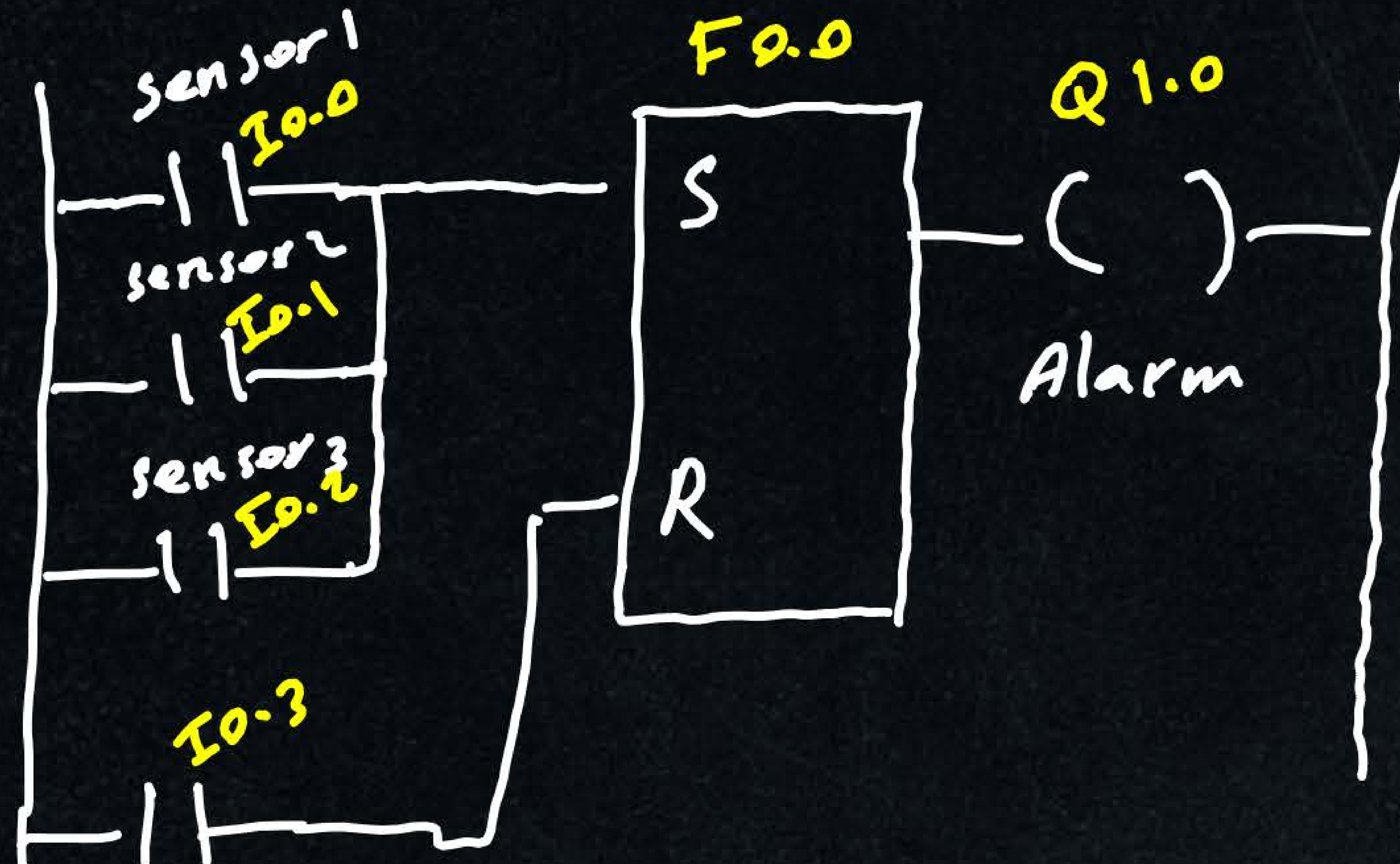
LD X400
OR Y430
ANI M100
OUT Y430
LD X401
OUT M100

Siemens



A I0.0
O Q1.0
AN F0.0
= Q1.0
A I0.1
= F0.0

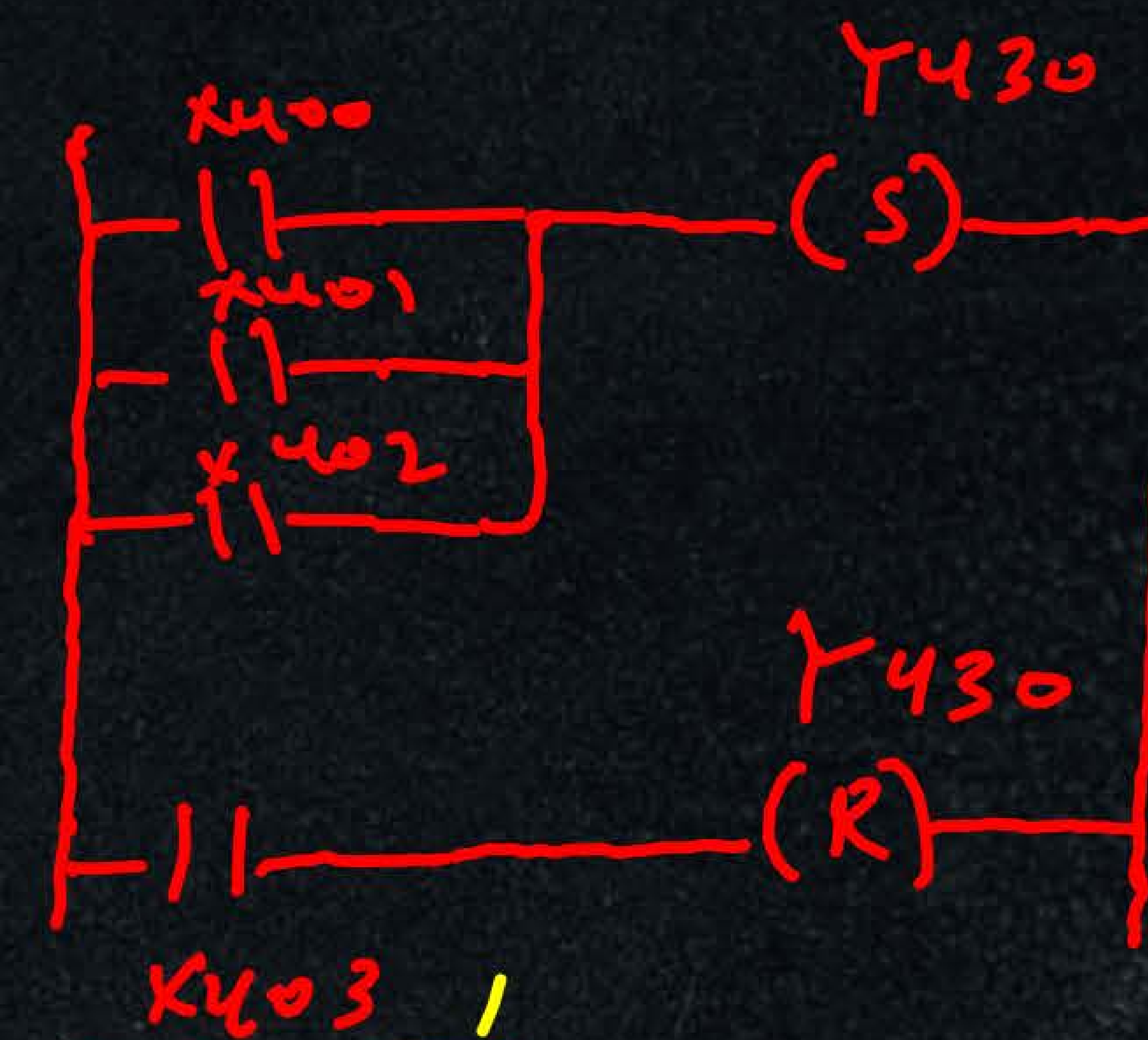
Ex Alarm system



clear
alarm
switch

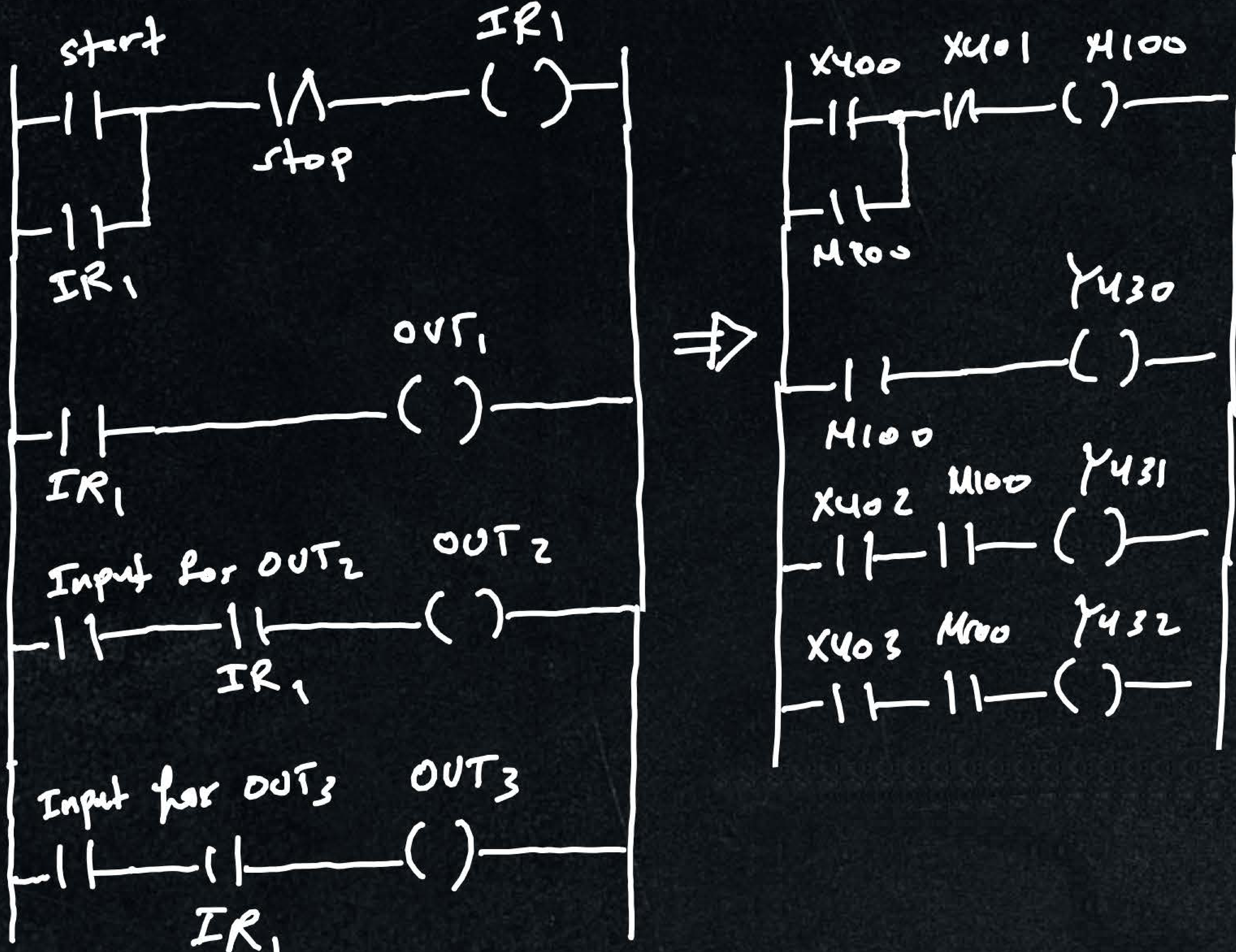
A I0.0
O I0.1
O I0.2
S F0.0
I0.3
R F0.0
A F0.0
= Q1.0

Mitsubishi



LD X400
OR X401
OR X402
S Y430
LD X403
R Y430

Ex Starting of Multiple outputs

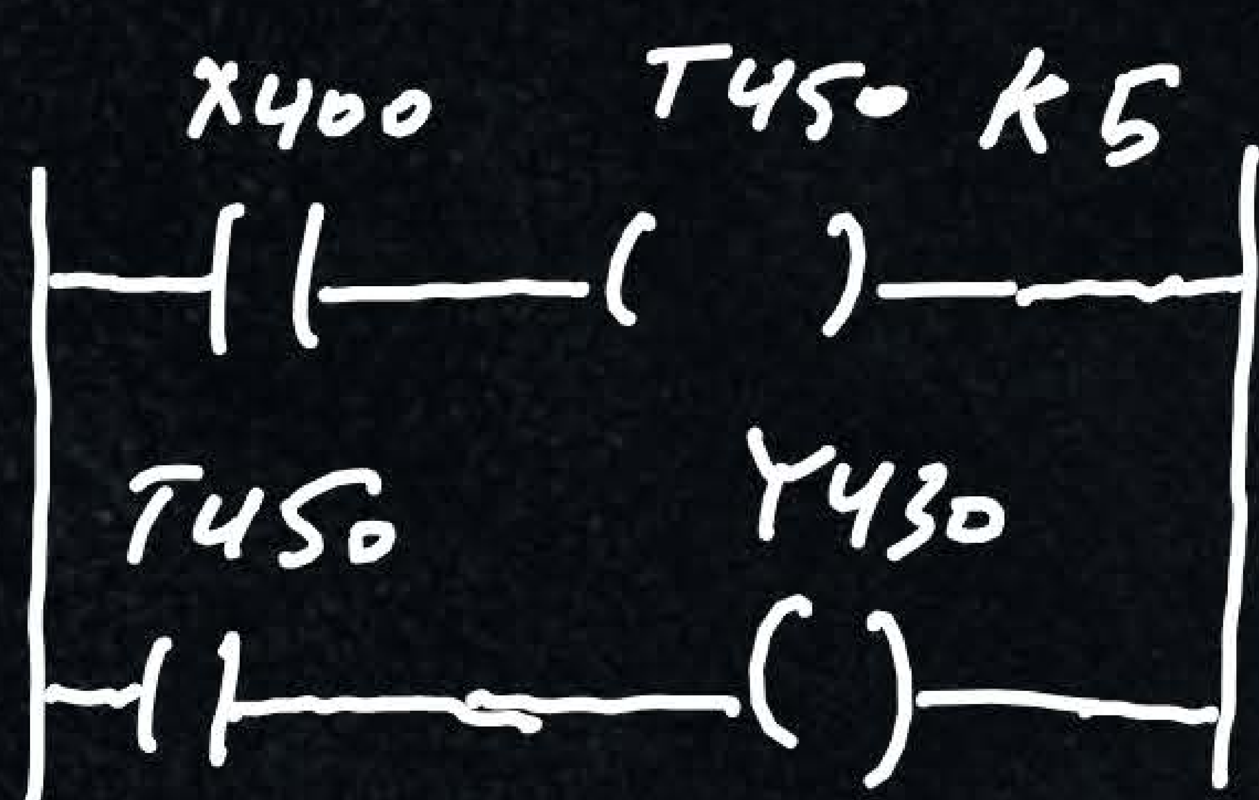



```
LD X400
OR M100
ANI X401
OUT M100
LD M100
OUT Y430
LD X402
AND M100
OUT Y431
LD X403
AND M100
OUT Y432
```

Timer

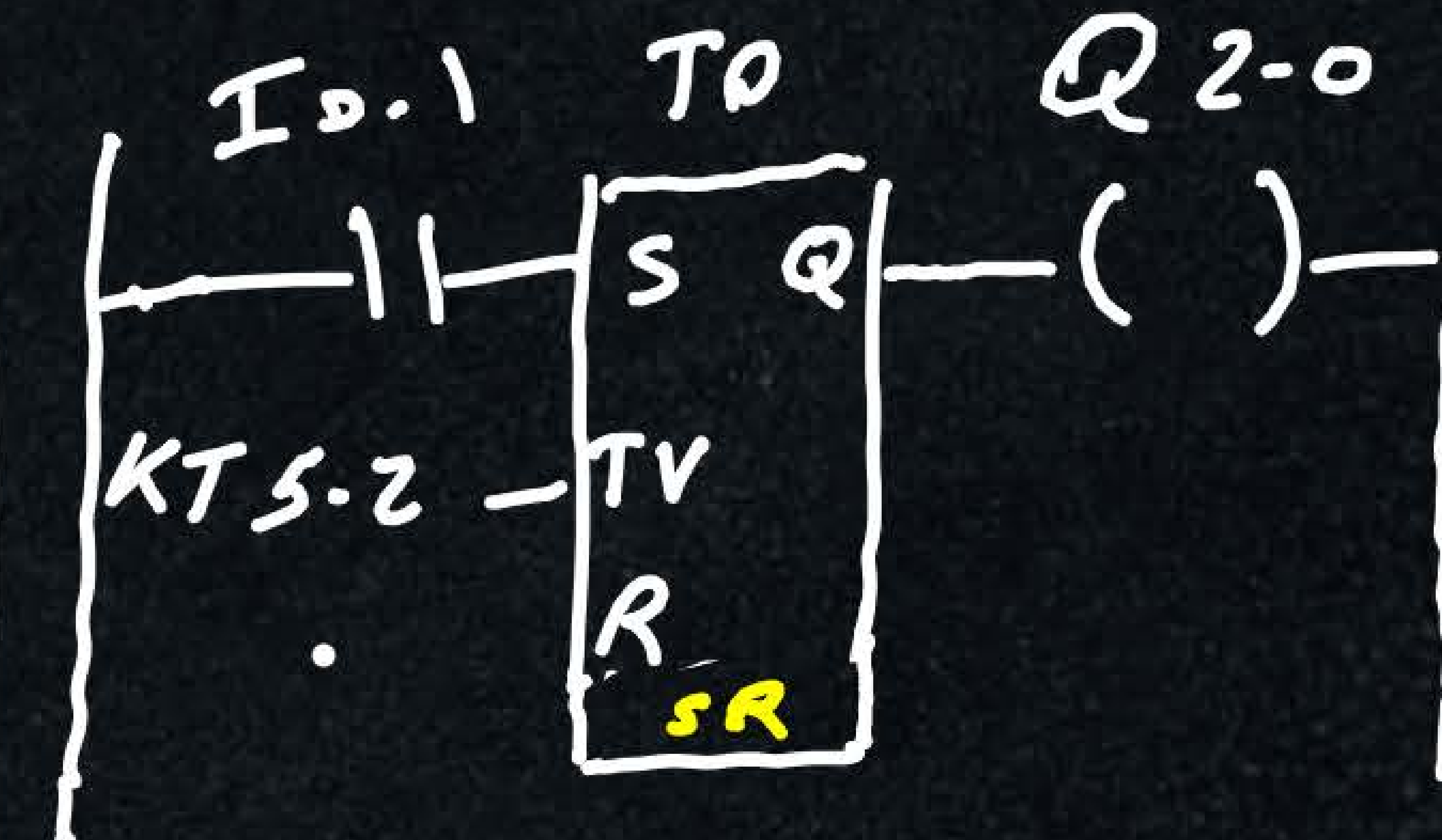
① on-delay timer

Mitsubishi



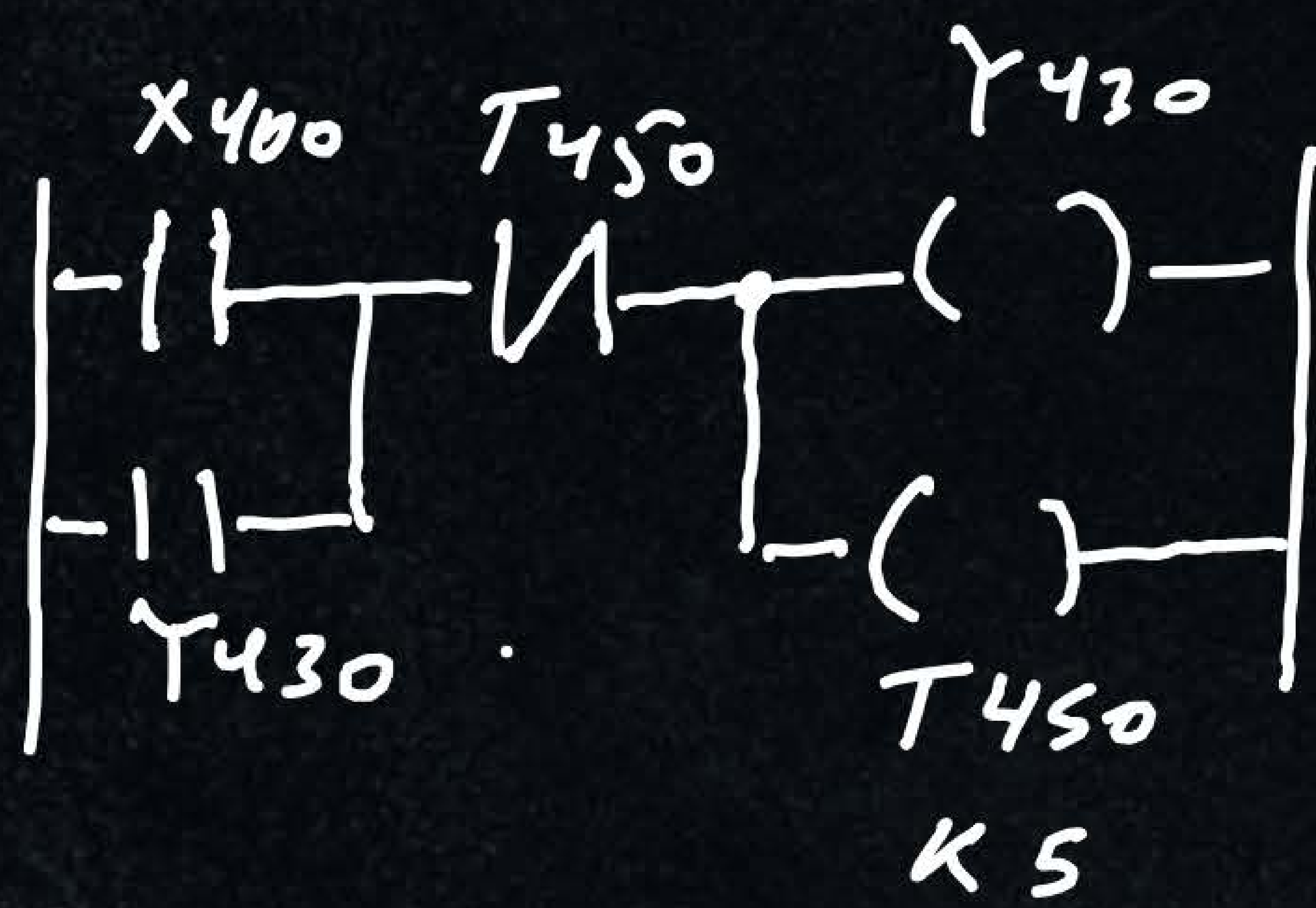

```
LD X400
OUT T453
 K5
LD T450
OUT Y430
```

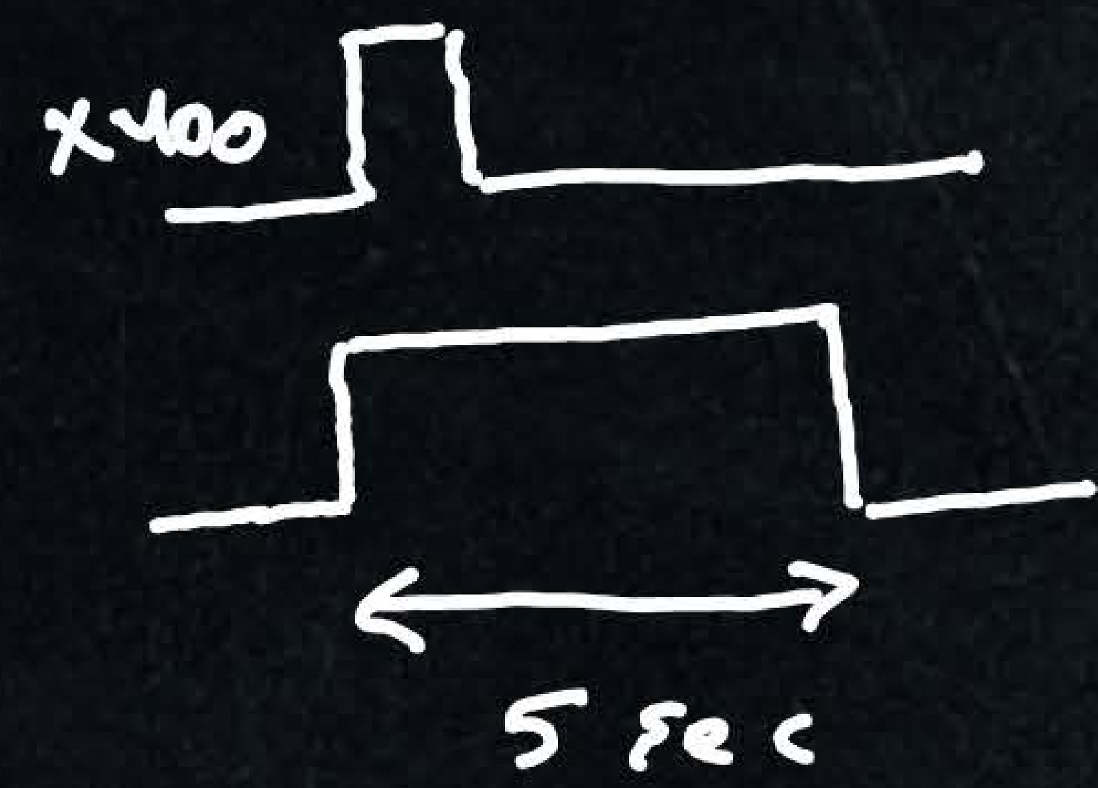
Siemens

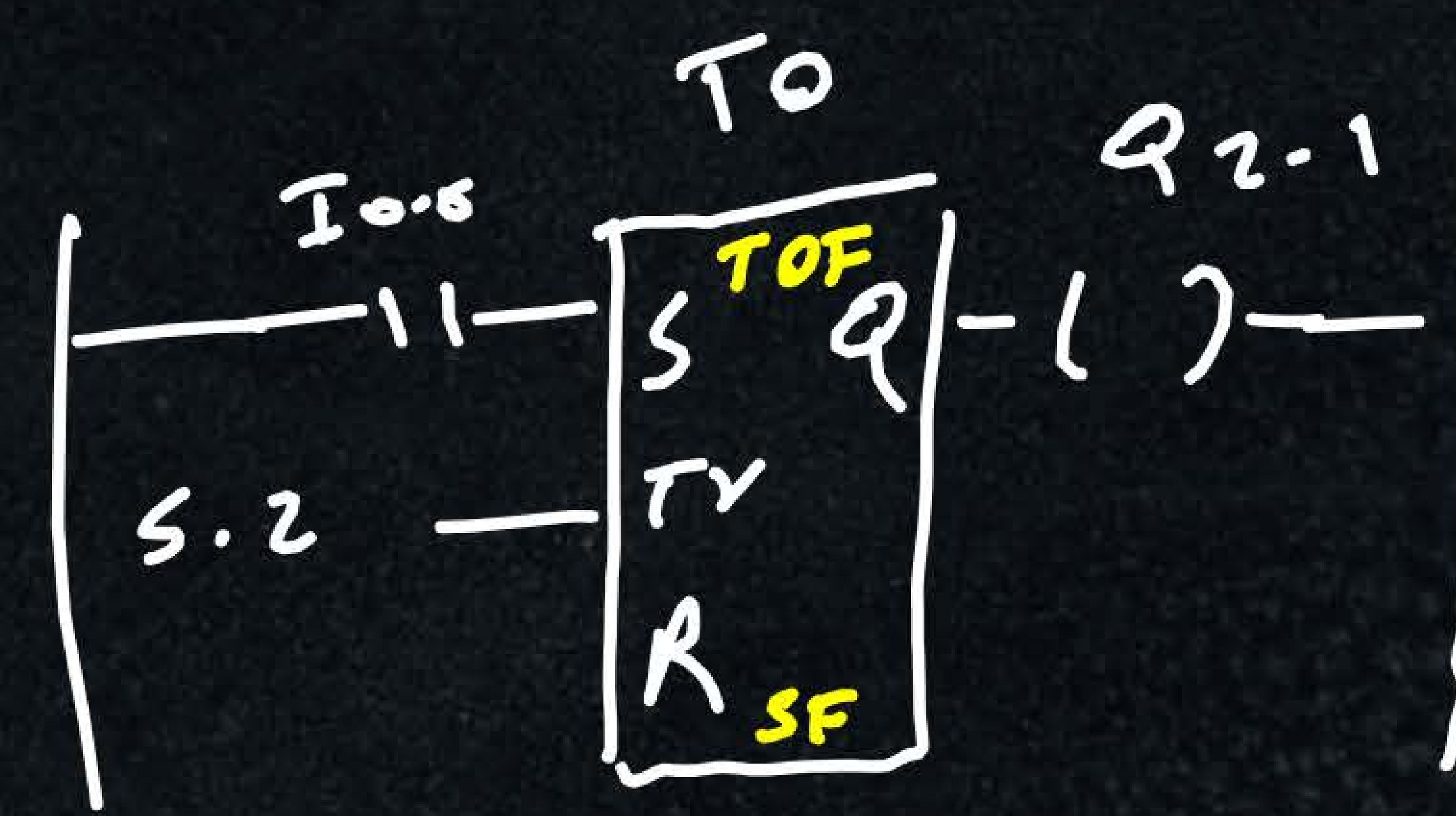



```
A I0.1
S R T0
LKT 5.2
A T0
= Q 2.0
```

② OFF-delay timer



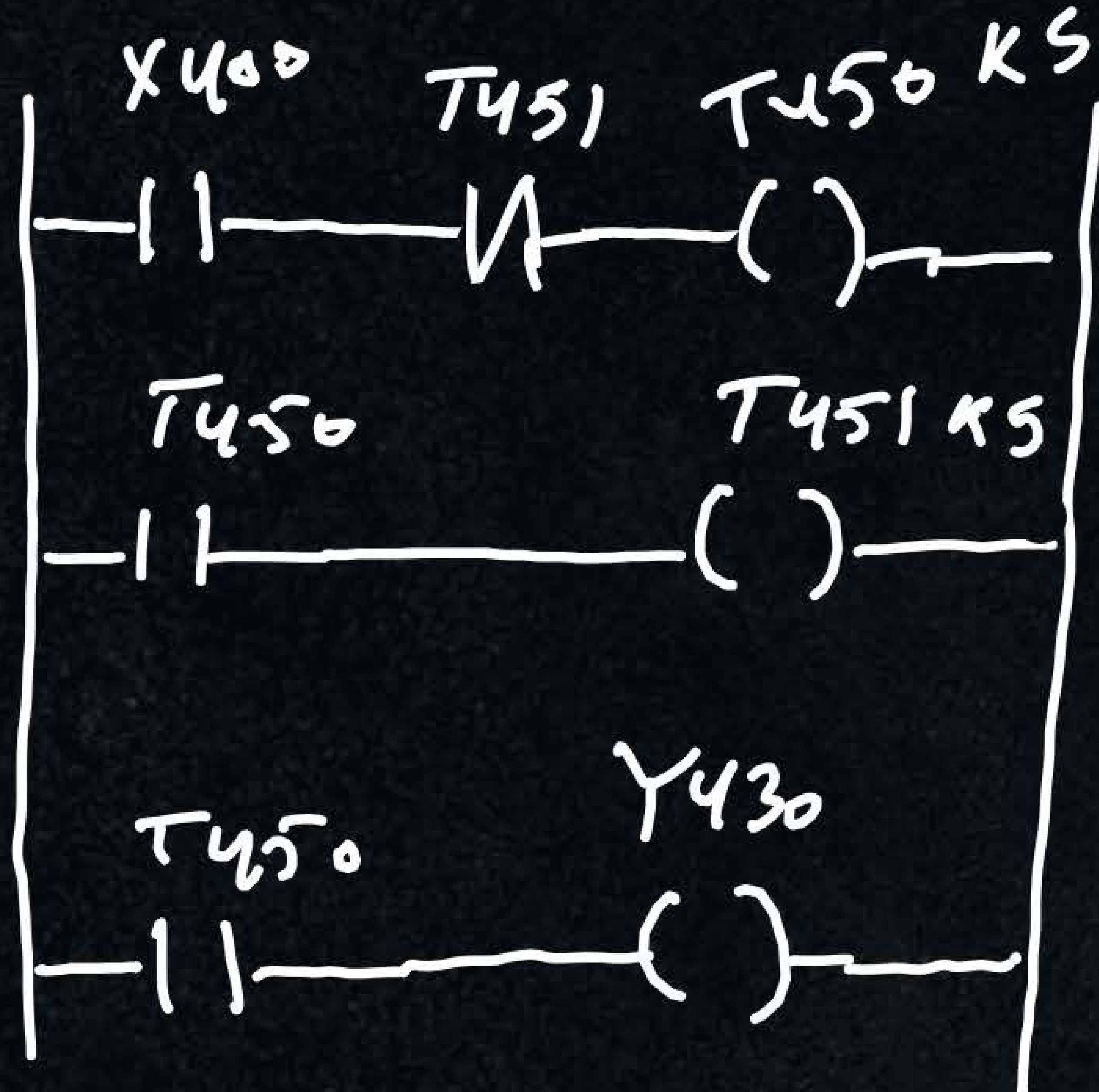


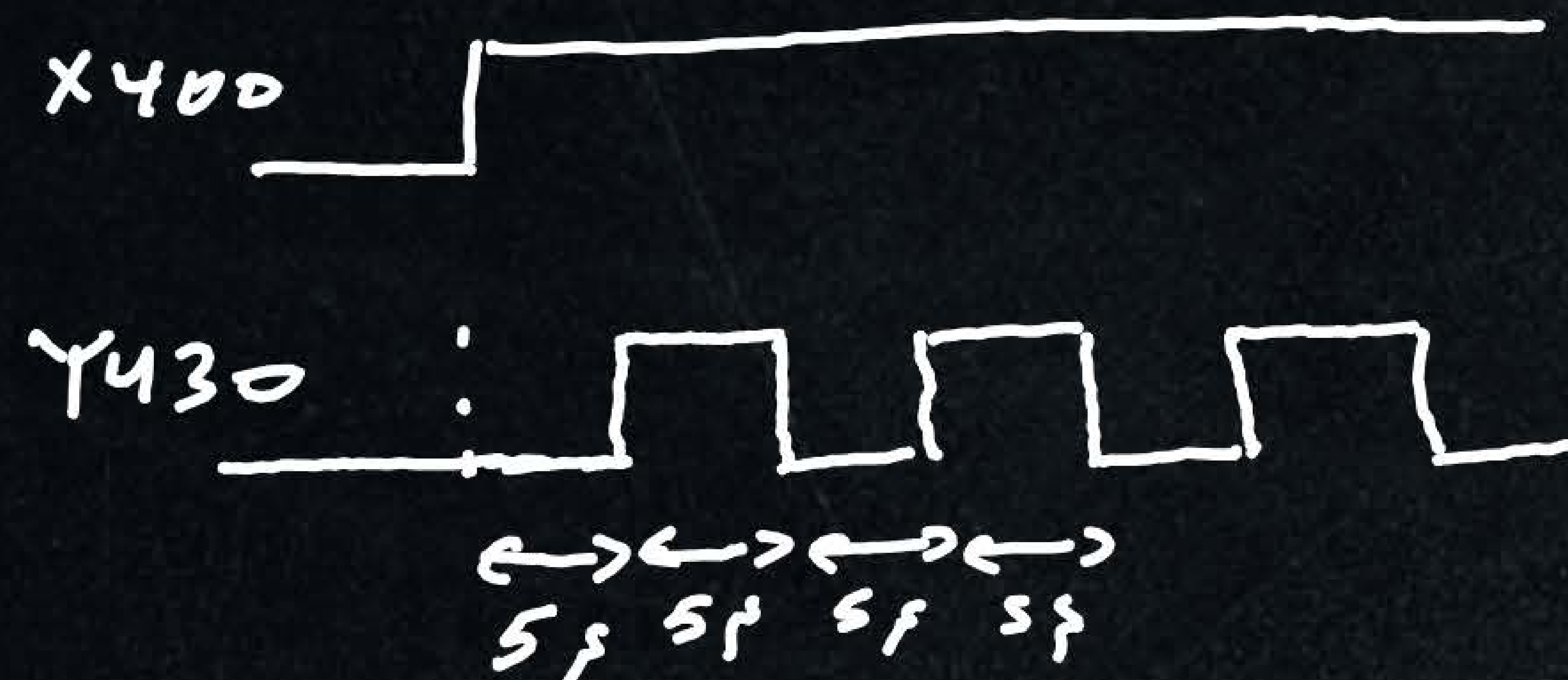



```
LD X400
OR Y430
ANI T450
OUT Y430
LD X400
OR Y430
ANI T450
OUT T450
    K 5
```

```
A I0.0
SF T0
LKT 5.2
A T0
= Q2.0
```

EX: ON/OFF cycle





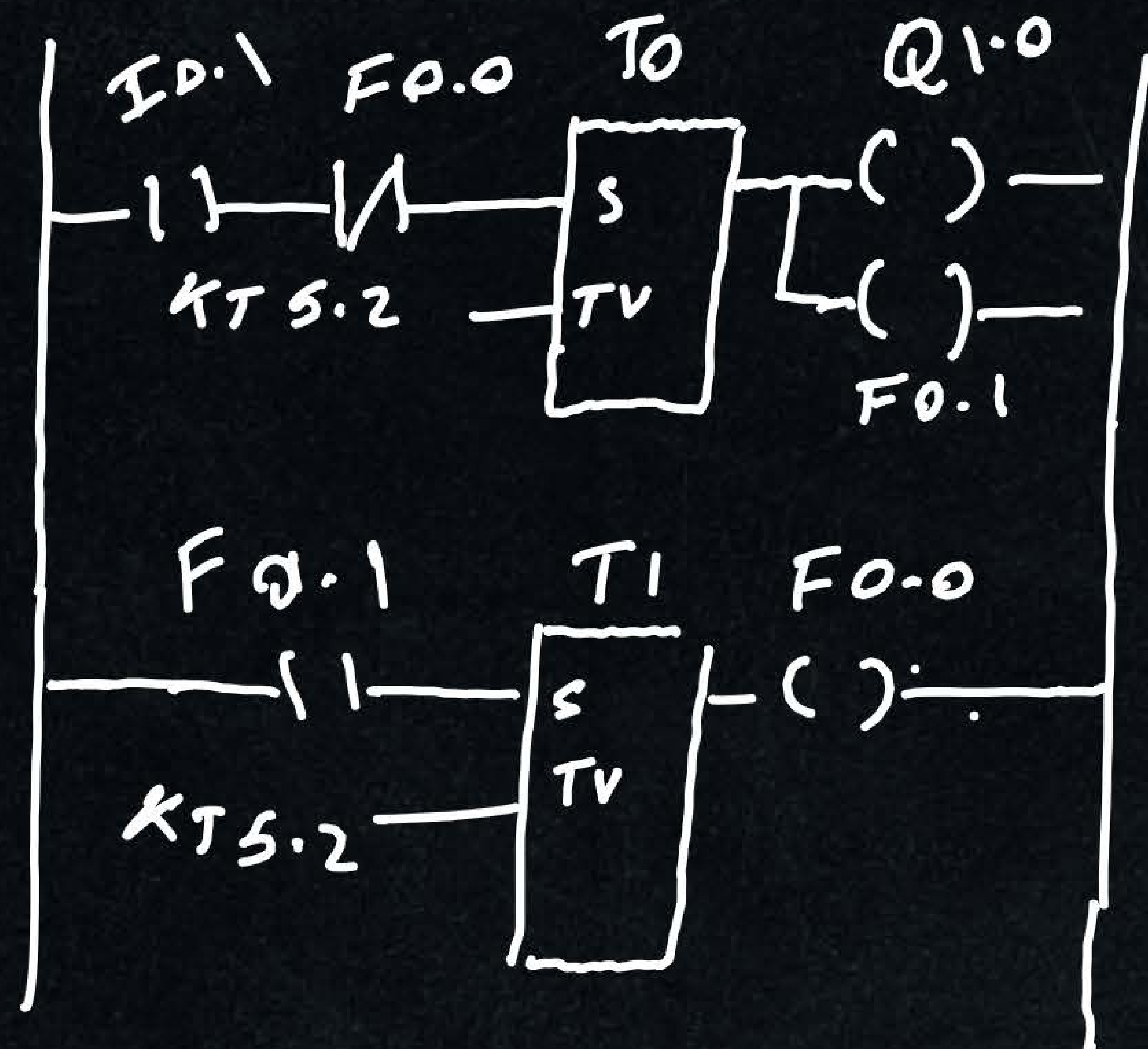
OFF 10 sec

ON 5 sec

```
LD X400
ANI T451
OUT T450
  K5
LD T450
OUT T451
  K5
```

```
LD T450
OUT Y430
```

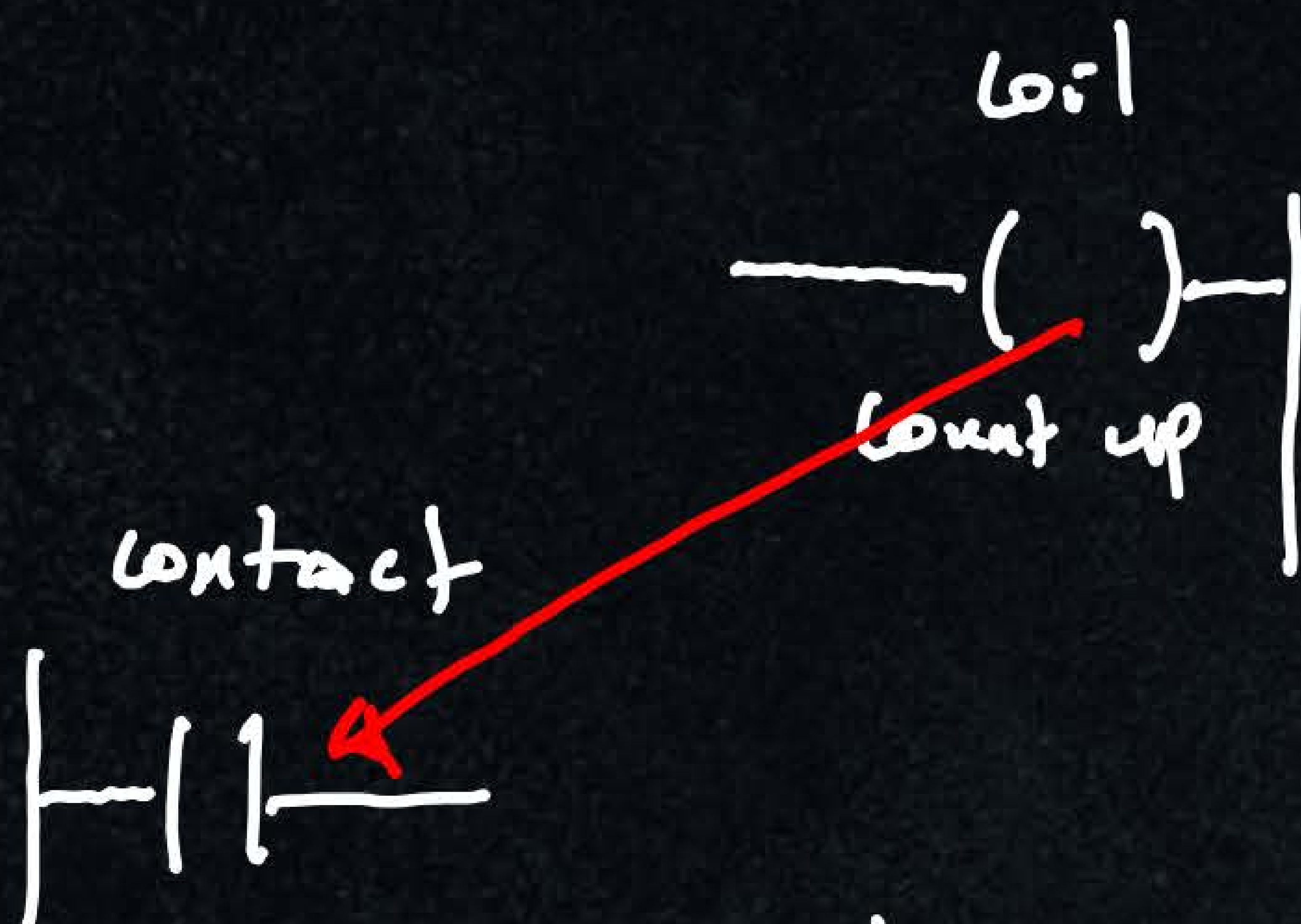
mitsubishi

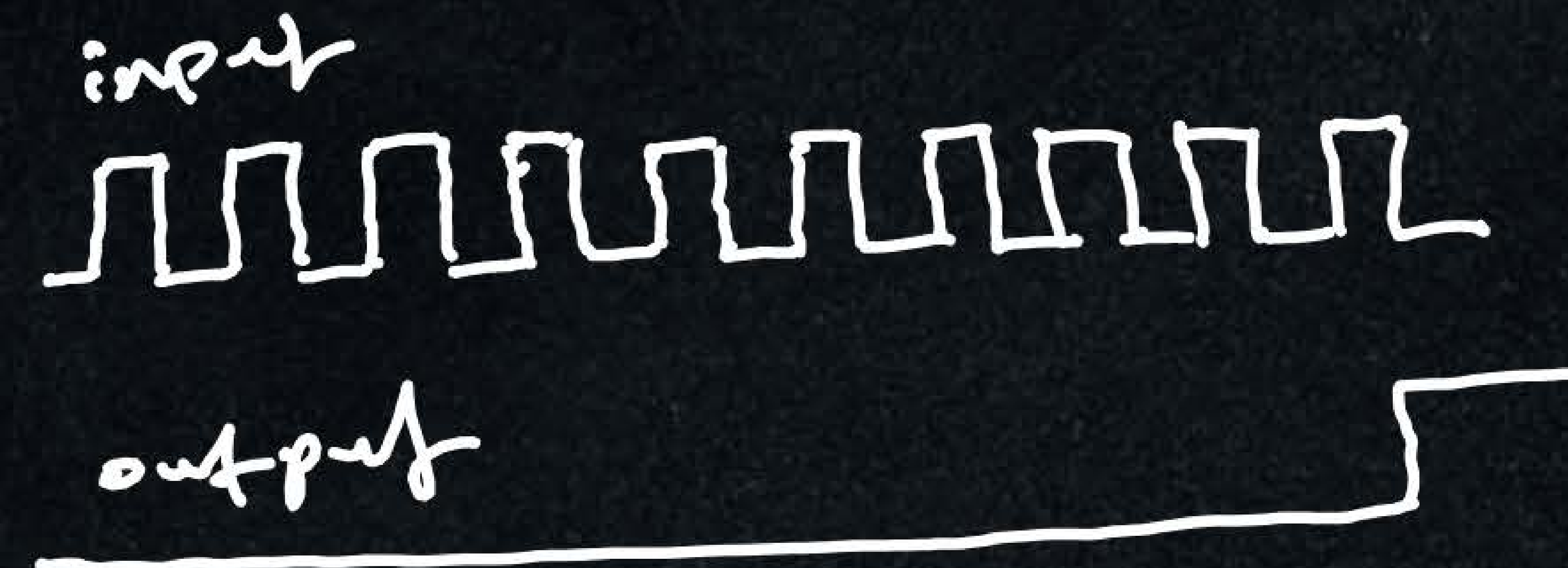

I0.1
F0.0
T0
Q1.0
S
KT5.2
TV
F0.1
F0.1
T1
F0.0
S
TV
KT5.2

Counters

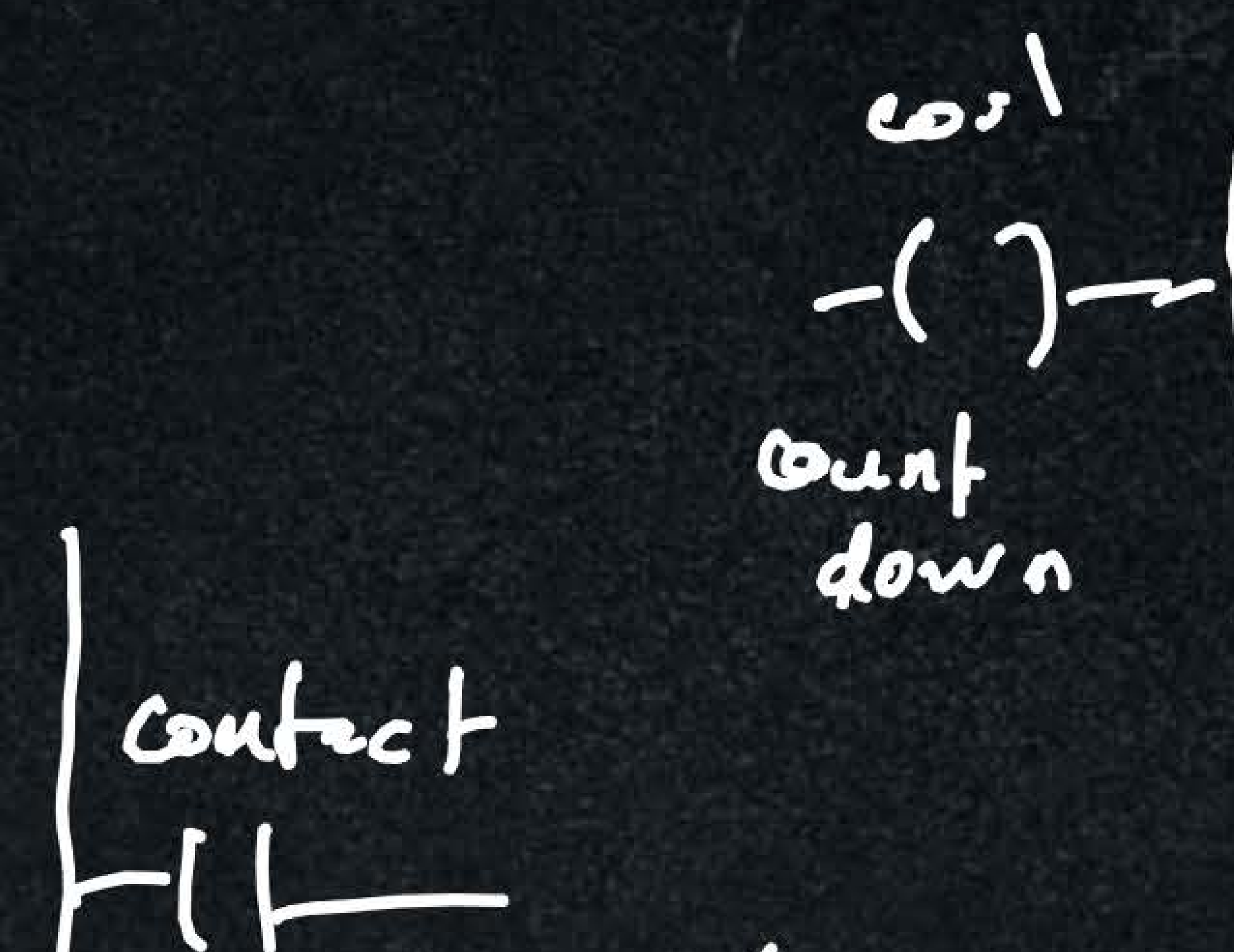
Up-counter → count from zero up to the preset value

Down-counter → count from present value to zero





Activated when set count reached after counting up from zero



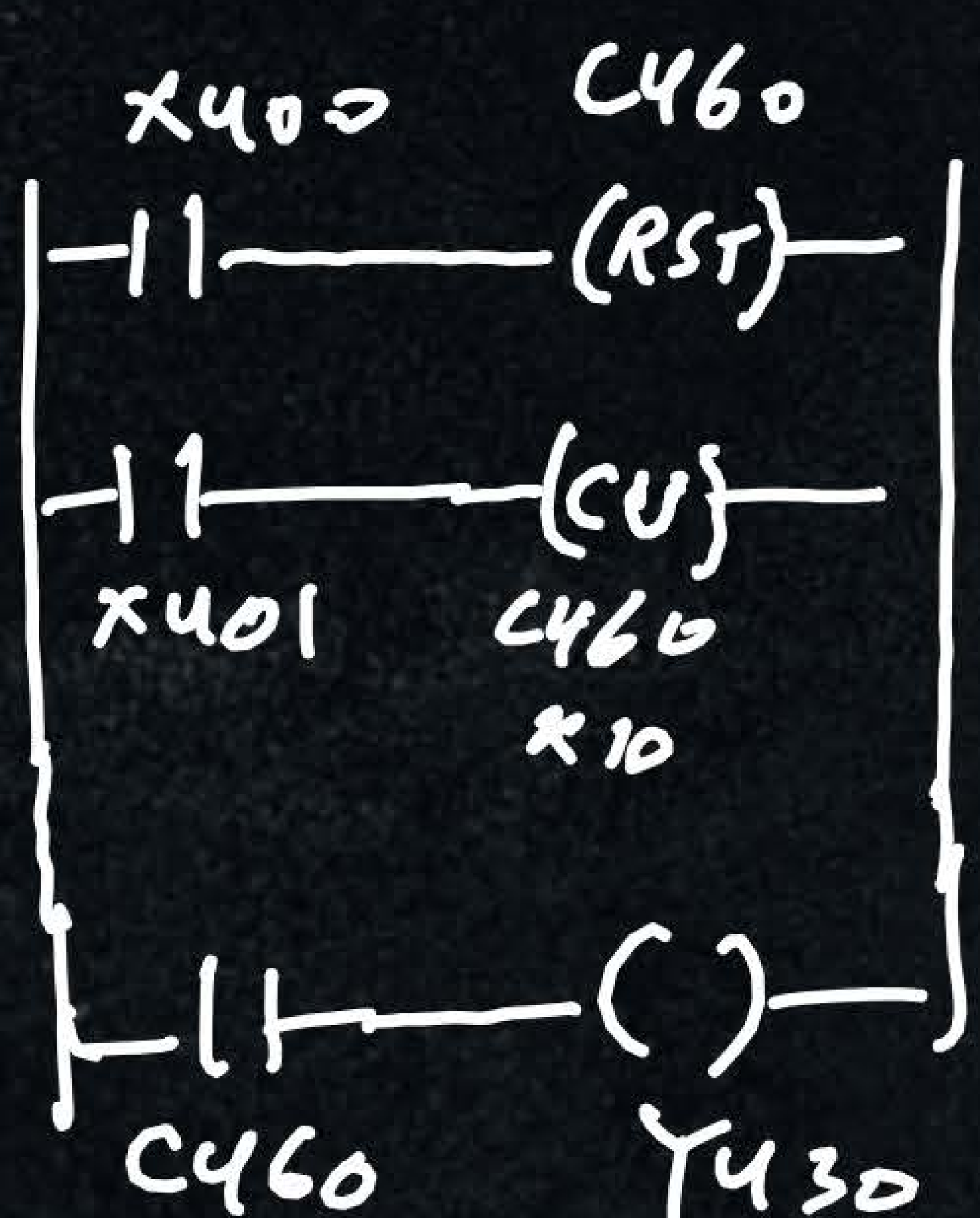
Activated when zero reached after counting down from preset value

Addresses

Mitsubishi C460, C461, ---

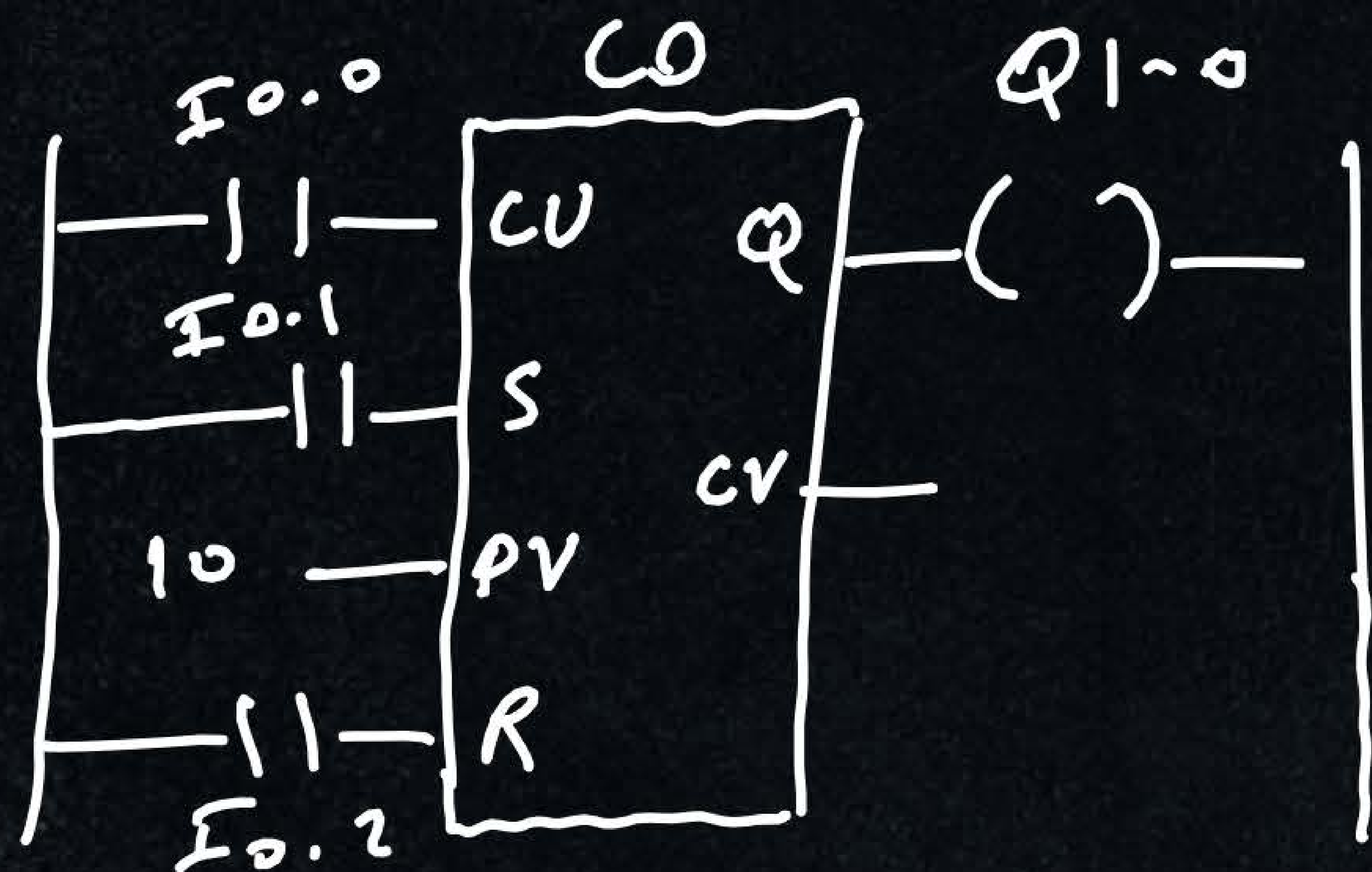
Siemens C0, C1, ----

Mitsubishi




```
LD X400
RST C460
LD X401
CU C460   or   OUT C460
   K10
LD C460
OUT Y430
```

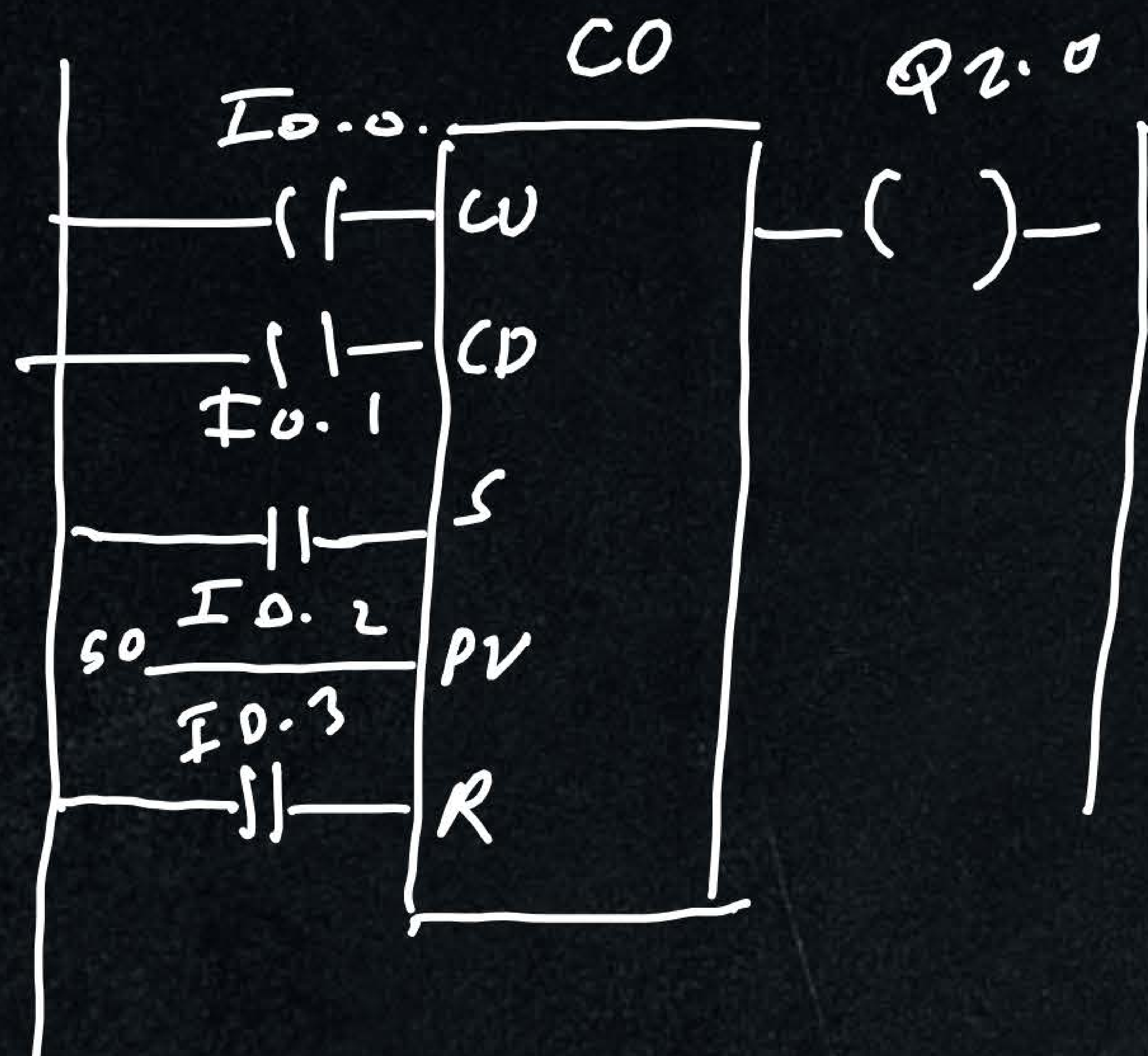
Siemens




```
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
LKC 50
A I 0.3
R C0
A C0
= Q2.0
```

Special Machines

Applications:

- printers
- Disk drives
- Robotics
- Tape drives
- ⋮

$\Rightarrow$ Contineous energy conversion is not required

Special Machines:

- stepper motors
- switched reluctance motors
- Servo motors

Stepper Motors

- They rotate by a specific number of degrees in response to an electrical pulse.
- Typical step sizes are 1.8°, 2°, 2.5°, 5°, 7.5°, & 15° for each pulse.
- Stepper motors well-suited for open loop position control.
- Such motors develop torques ranging from 1 μN.m upto 40 N.m

  1 μN.m ↓ tiny wrist watch motor

  40 N.m ↓ machine tool appli.

- Types of stepper motors

1) Variable Reluctance (VR)
2) permanent Magnent (PM)
3) Hybrid.

① VR stepper motor

* It is constructed from ferromagnetic material with salient poles. The stator is made from a stack of steel laminations and has a number of equally spaced projecting poles (teeth) each wound with an exciting coil. The rotor has teeth of the same width as the stator poles.

- step angle $= \alpha = \dfrac{N_s - N_r}{N_s N_r} \times 360°$

  ↓

  angle, which the rotor moves it per each pulse

  N_s : stator poles

  N_r : stator teeth

or $\alpha = \dfrac{360°}{m_s N_r}$; where m_s is the number of stator phases.

- Resolution = Number of steps / revolution $= \dfrac{360°}{\alpha}$

- If f is the stepping rate or stepping frequency or pulse rate in pulses per second (PPS), then the motor's speed is:

$$n = \frac{\alpha}{360°} \times f \quad [rpm]$$

Drive circuit of VR stepper motor