PROGRAMMABLE LOGIC CONTROLLERS

LAB 3

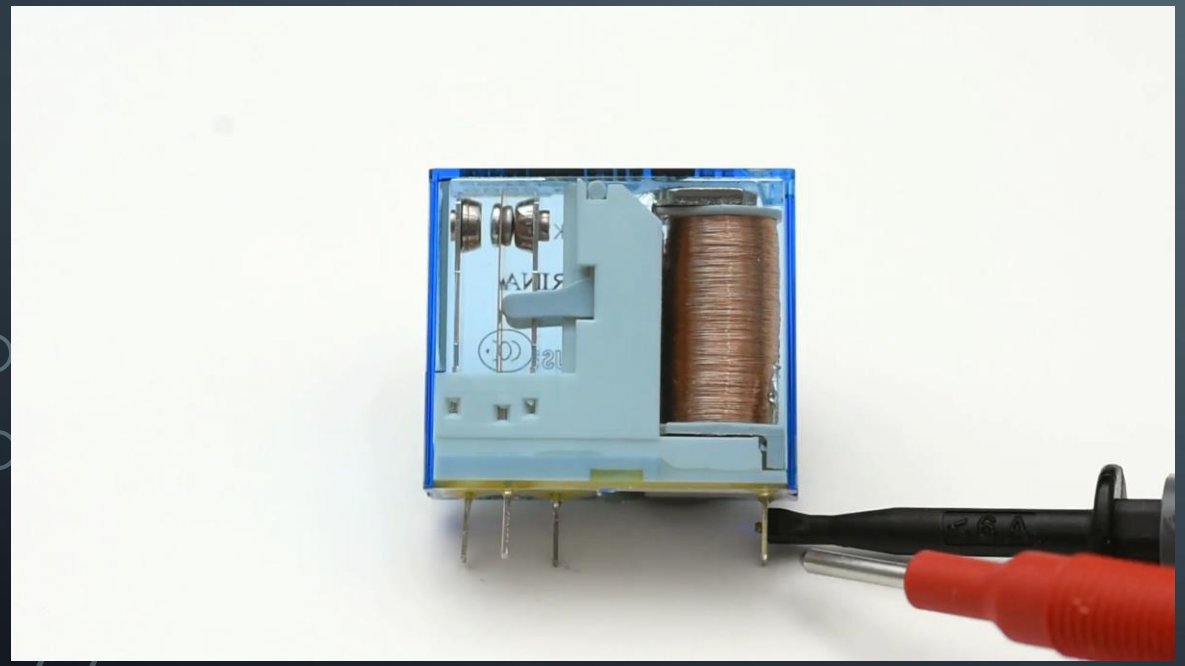
SCHEDULE

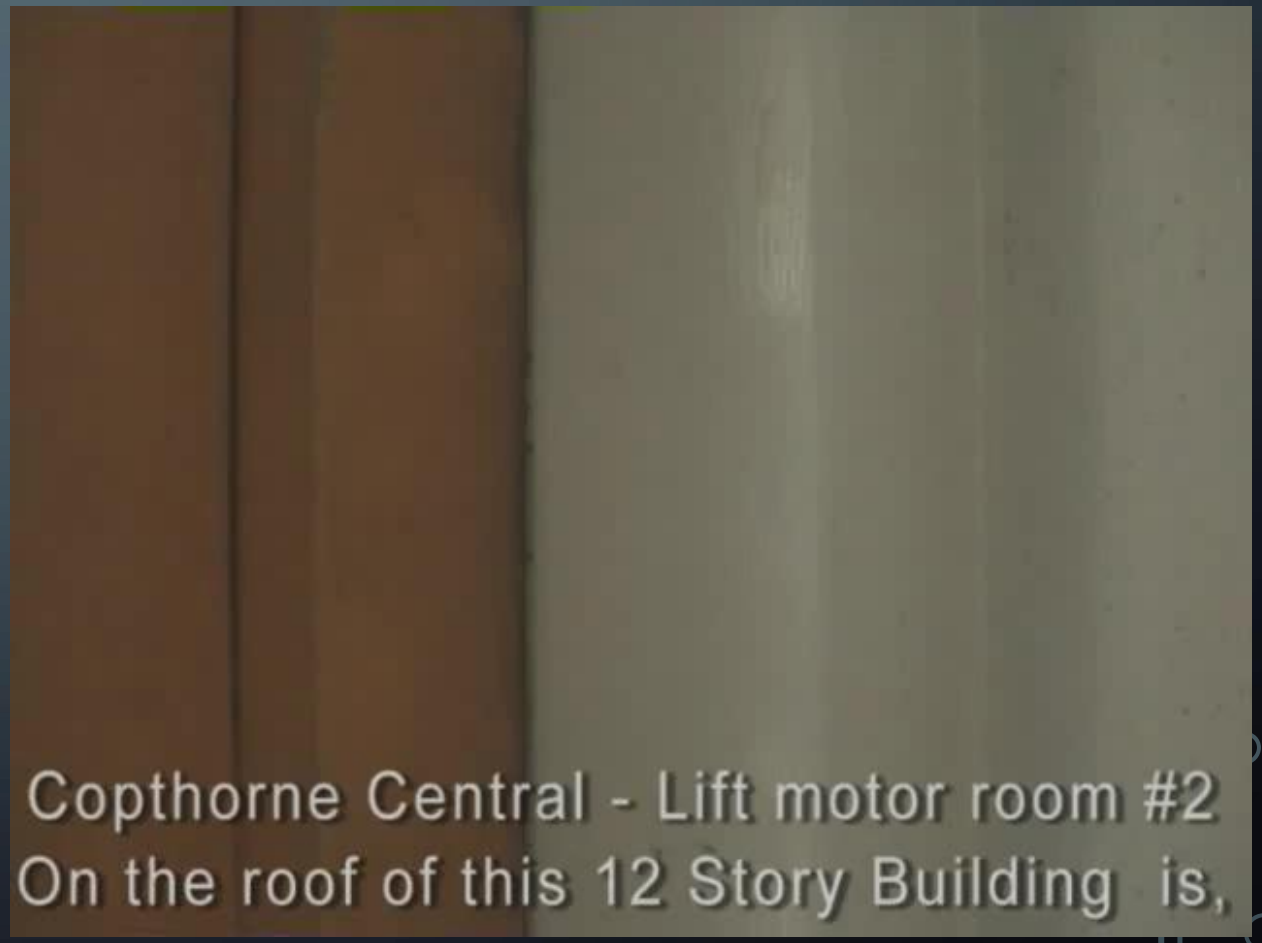
3	September 16 Forced and Free Vibrations Part II Gear Systems	September 16, 18 Forced and Free Vibrations Part II Gear Systems
4	September 23 No Lecture	September 23, 27 Forced and Free Vibrations Part II Gear Systems
5	September 30 Exam I PLC Tank	September 30, October 16 PLC Tank
6	October 7 No Lecture	October 9, 11 PLC Tank
7	October 14 Control Theory Unstable Systems	October 14, 18 Control Theory Unstable Systems

OUTLINE

- Intro to PLC
- Fundamentals of Logic
- Boolean algebra
- Ladder Logic
- Lab Procedures

RELAY CONTROLLED LOGIC





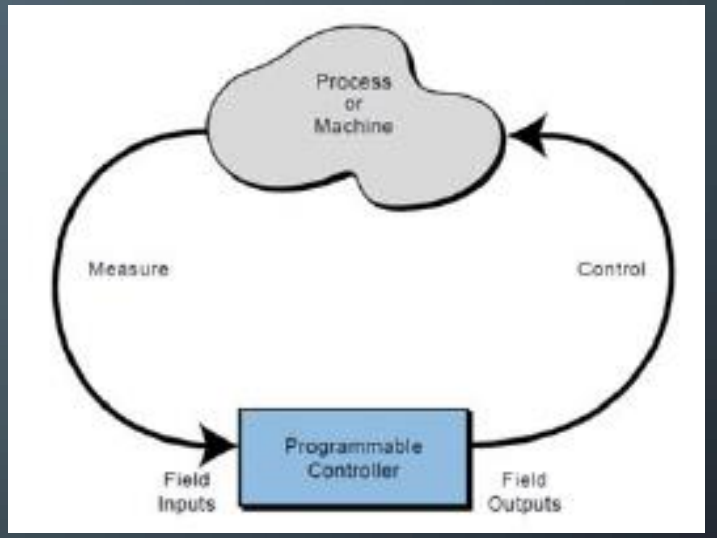
PLC

vs

Relay Logic

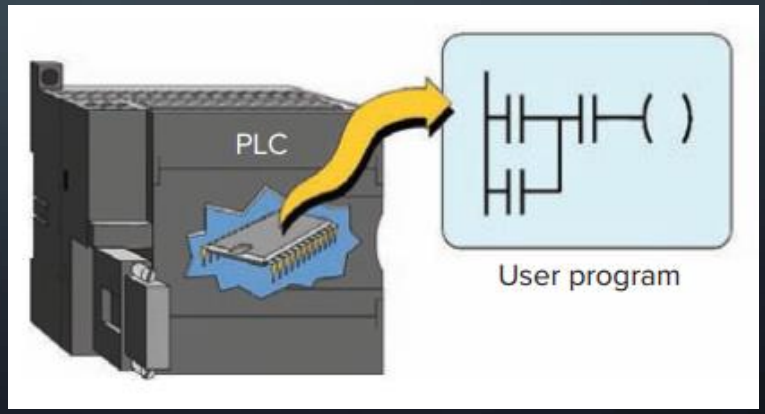
PROGRAMMABLE LOGIC CONTROLLERS

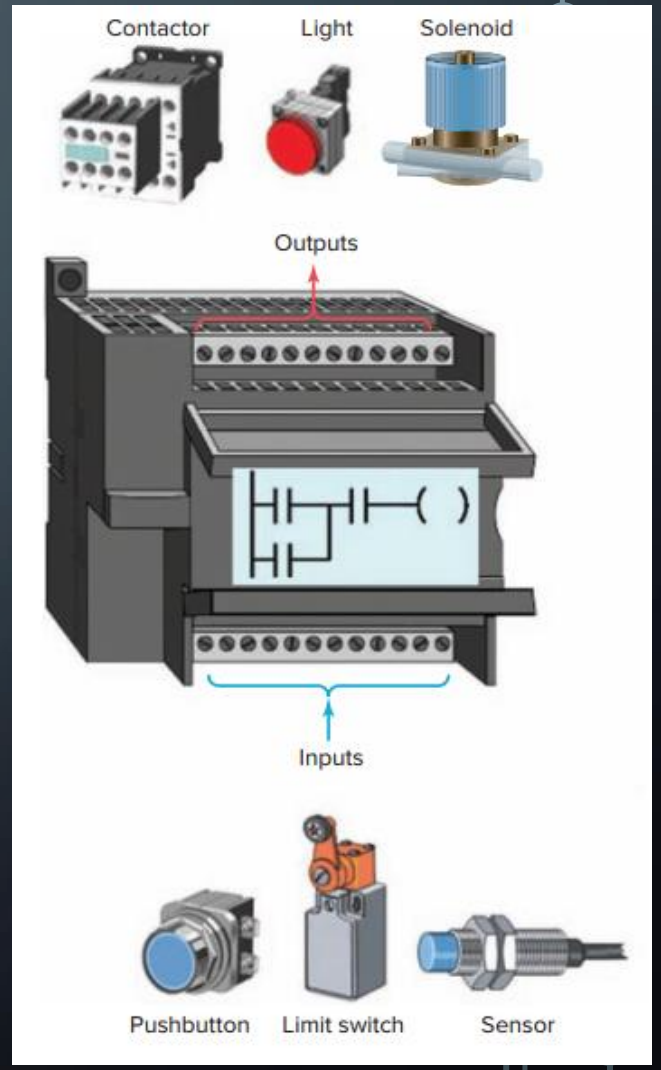
- PLC: real-time system
- Replace relay logic
- Functions
  - Relay switch
  - Timing
  - Counting
  - Signal processing



PROGRAMMABLE LOGIC CONTROLLERS

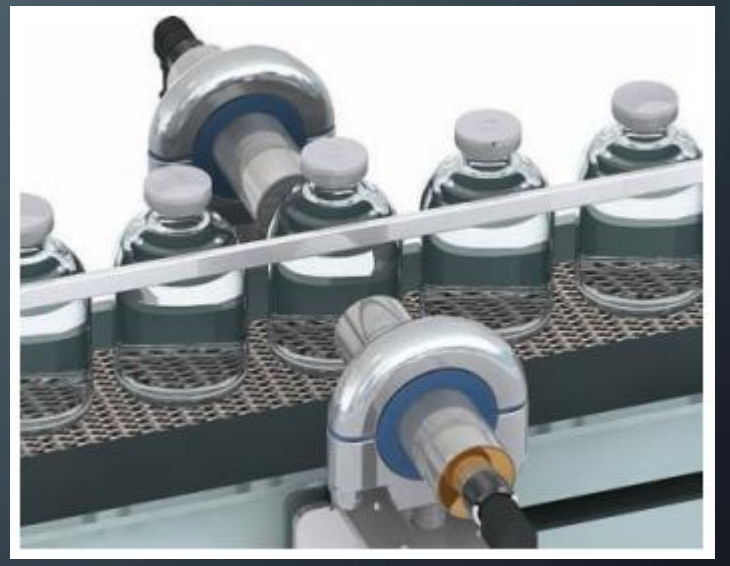
- Benefits
  - Increased Reliability
  - More Flexibility
  - Lower Cost
  - Communications Capability
  - Faster Response Time
  - Easier to Troubleshoot
  - Easier to Test Field Devices

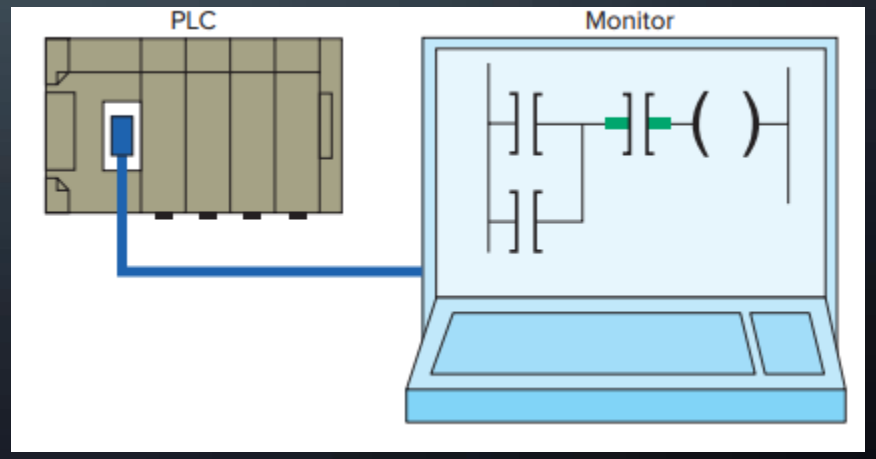




PROGRAMMABLE LOGIC CONTROLLERS

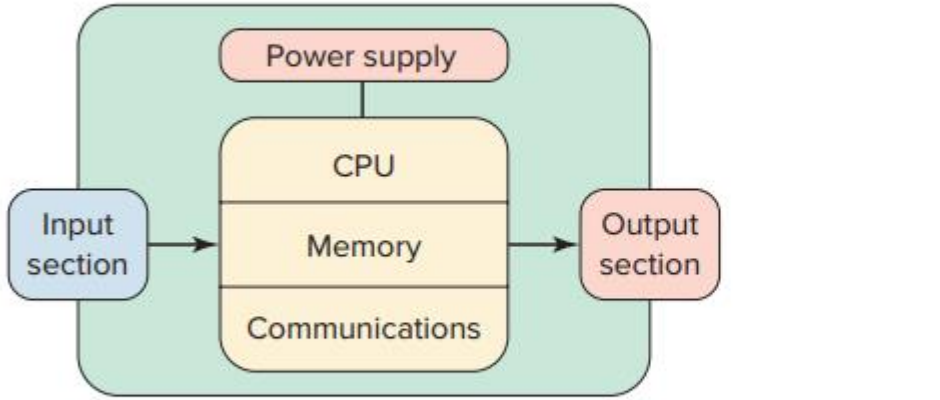
- Benefits
  - Increased Reliability
  - More Flexibility
  - Lower Cost
  - Communications Capability
  - Faster Response Time
  - Easier to Troubleshoot
  - Easier to Test Field Devices



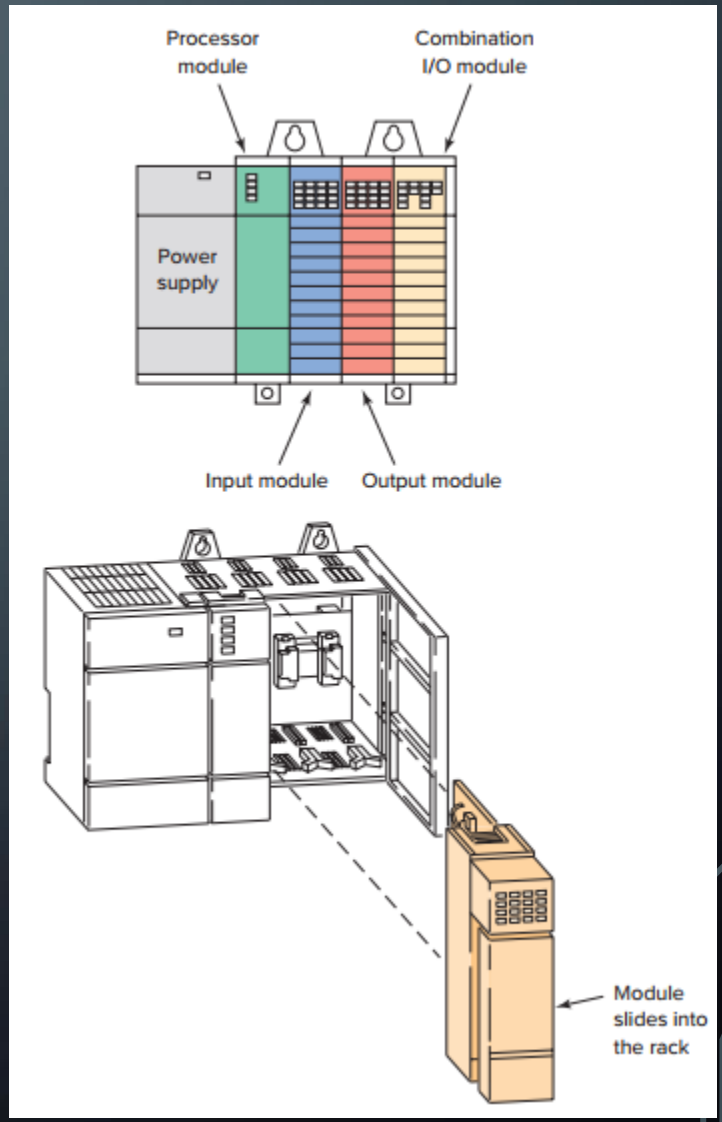


PARTS OF PLC

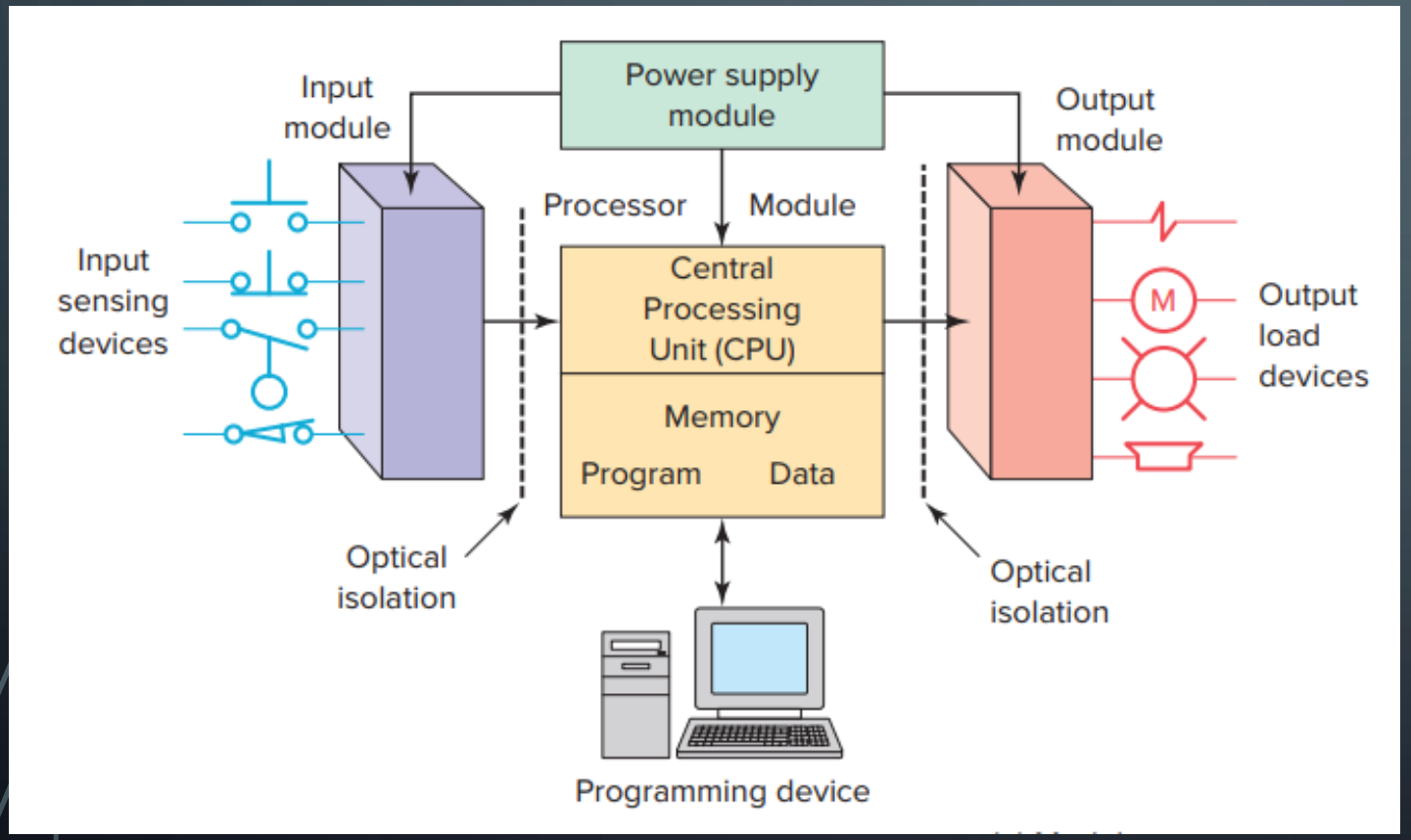
- Fixed I/O vs Modular I/O

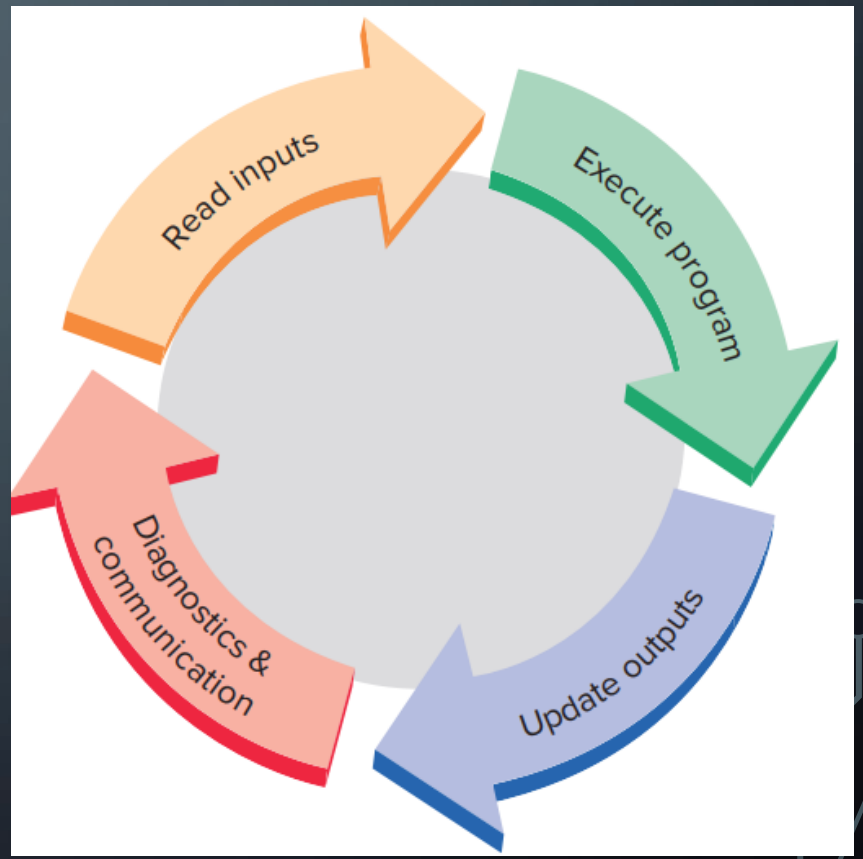


(*b*) Fixed type

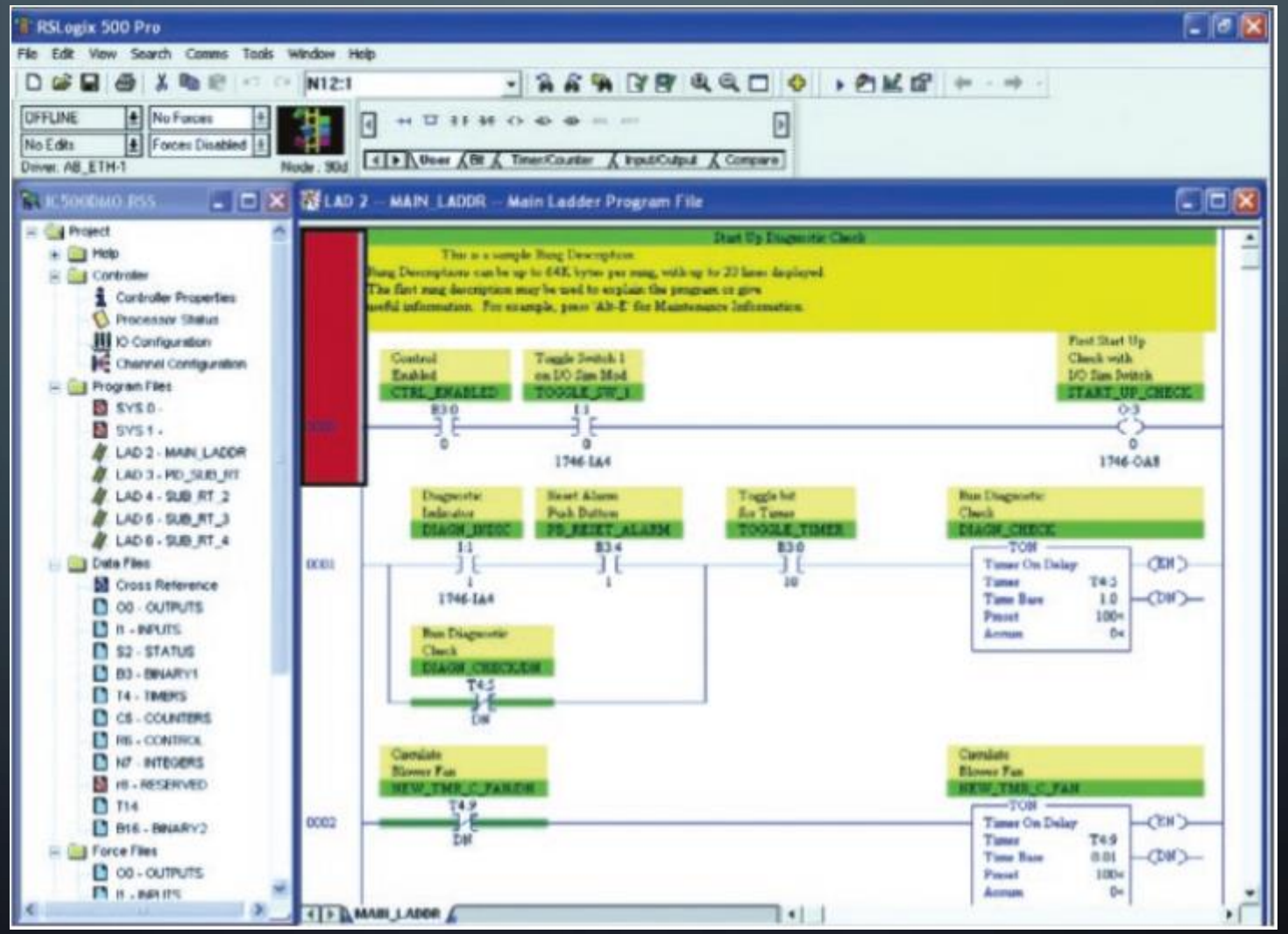


PARTS OF PLC


Input module
Power supply module
Output module
Processor Module
Input sensing devices
Central Processing Unit (CPU)
Memory
Program
Data
M
Output load devices
Optical isolation
Optical isolation
Programming device

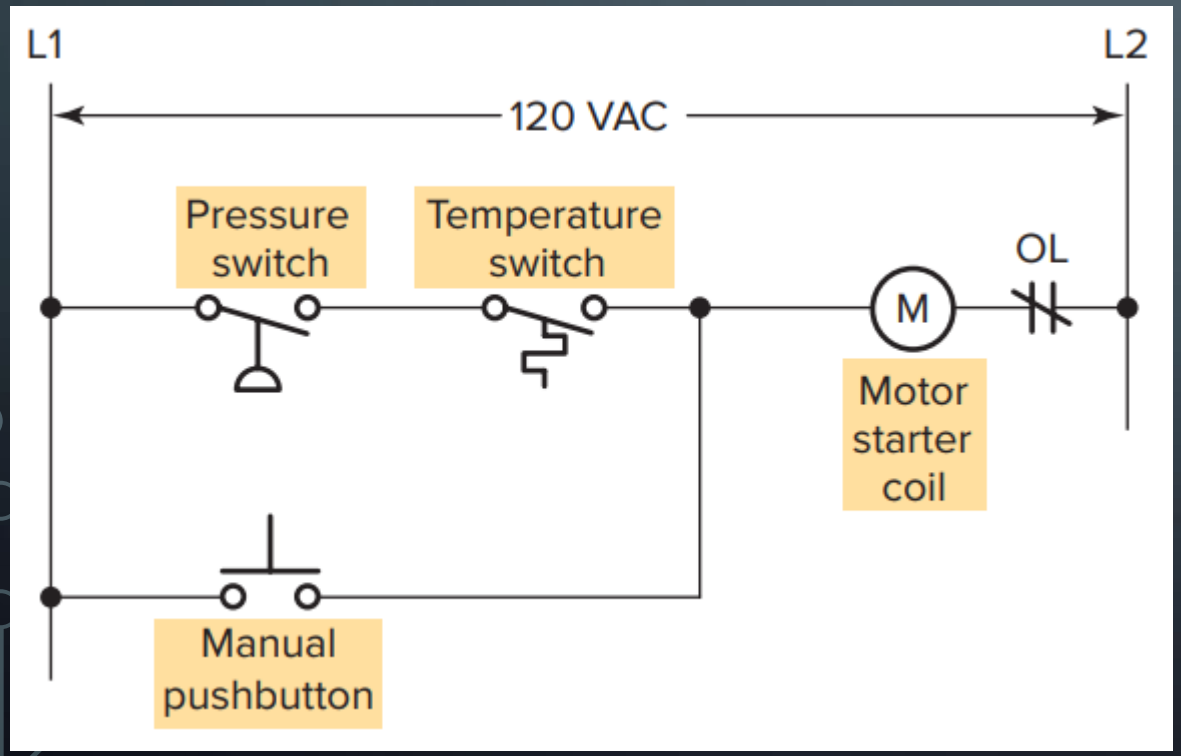

Read inputs
Execute program
Update outputs
Diagnostics & communication

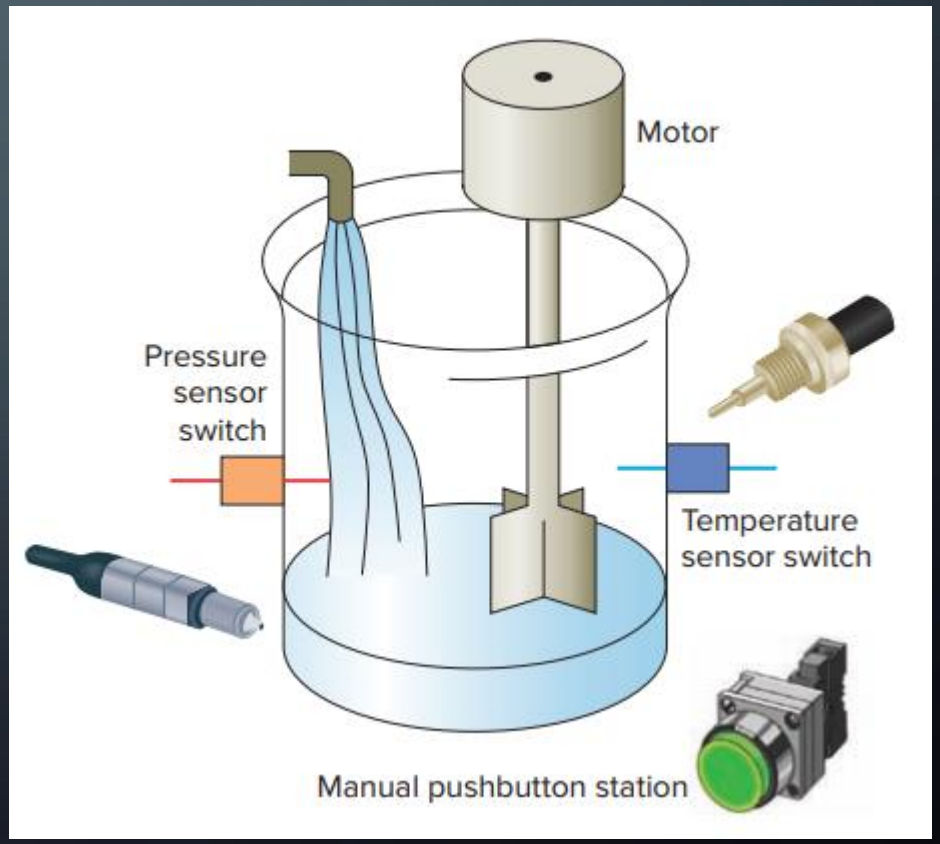
PARTS OF PLC


RSLogix 500 Pro
File Edit View Search Comms Tools Window Help
N12:1
OFFLINE
No Forces
No Edits
Forces Disabled
Driver: AB_ETH-1
Node: 30d
Project
Help
Controller
Controller Properties
Processor Status
IO Configuration
Channel Configuration
Program Files
SYS 0 -
SYS 1 -
LAD 2 - MAIN_LADDR
LAD 3 - PID_SUB_RT
LAD 4 - SUB_RT_2
LAD 5 - SUB_RT_3
LAD 6 - SUB_RT_4
Data Files
Cross Reference
O0 - OUTPUTS
I1 - INPUTS
S2 - STATUS
B3 - BINARY1
T4 - TIMERS
C5 - COUNTERS
R6 - CONTROL
N7 - INTEGERS
T14
B16 - BINARY2
Force Files
O0 - OUTPUTS
LAD 2 -- MAIN_LADDR -- Main Ladder Program File
Start Up Diagnostic Check
Control Enabled
CTRL_ENABLED
Toggle Switch 1 on I/O Jim Mod
TOGGLE_SW_1
First Start Up Check with I/O Jim Switch
START_UP_CHECK
1746-IA4
1746-OA8
Diagnostic Indicator
DIAGN_INDIC
Reset Alarm Push Button
PB_RESET_ALARM
Toggle bit for Timer
TOGGLE_TIMER
Run Diagnostic Check
DIAGN_CHECK
TON
Timer On Delay
Timer
Time Base
Preset
Accum
EN
DN
Run Diagnostic Check
DIAGN_CHECK_RUN
Circulate Blower Fan
NEW_TMR_C_FANDN
NEW_TMR_C_FAN
MAIN_LADDR

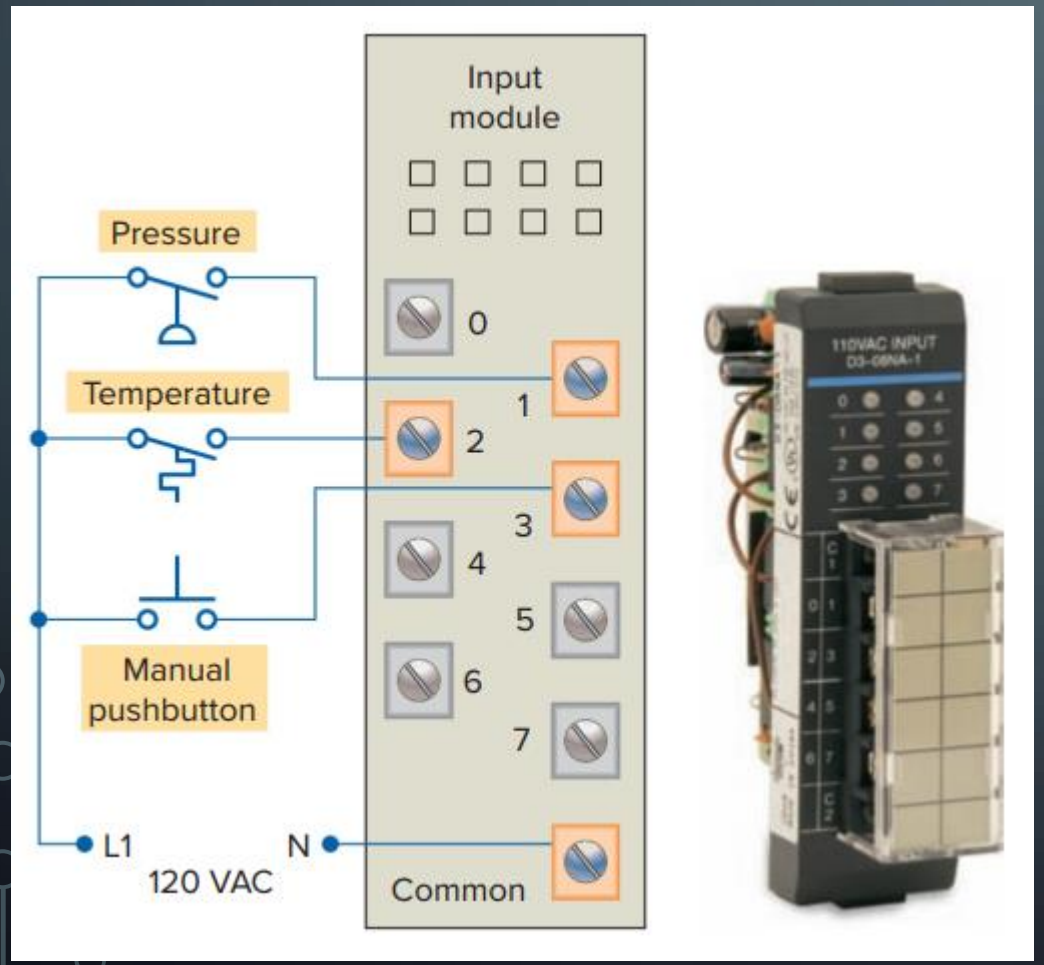
AN EXAMPLE

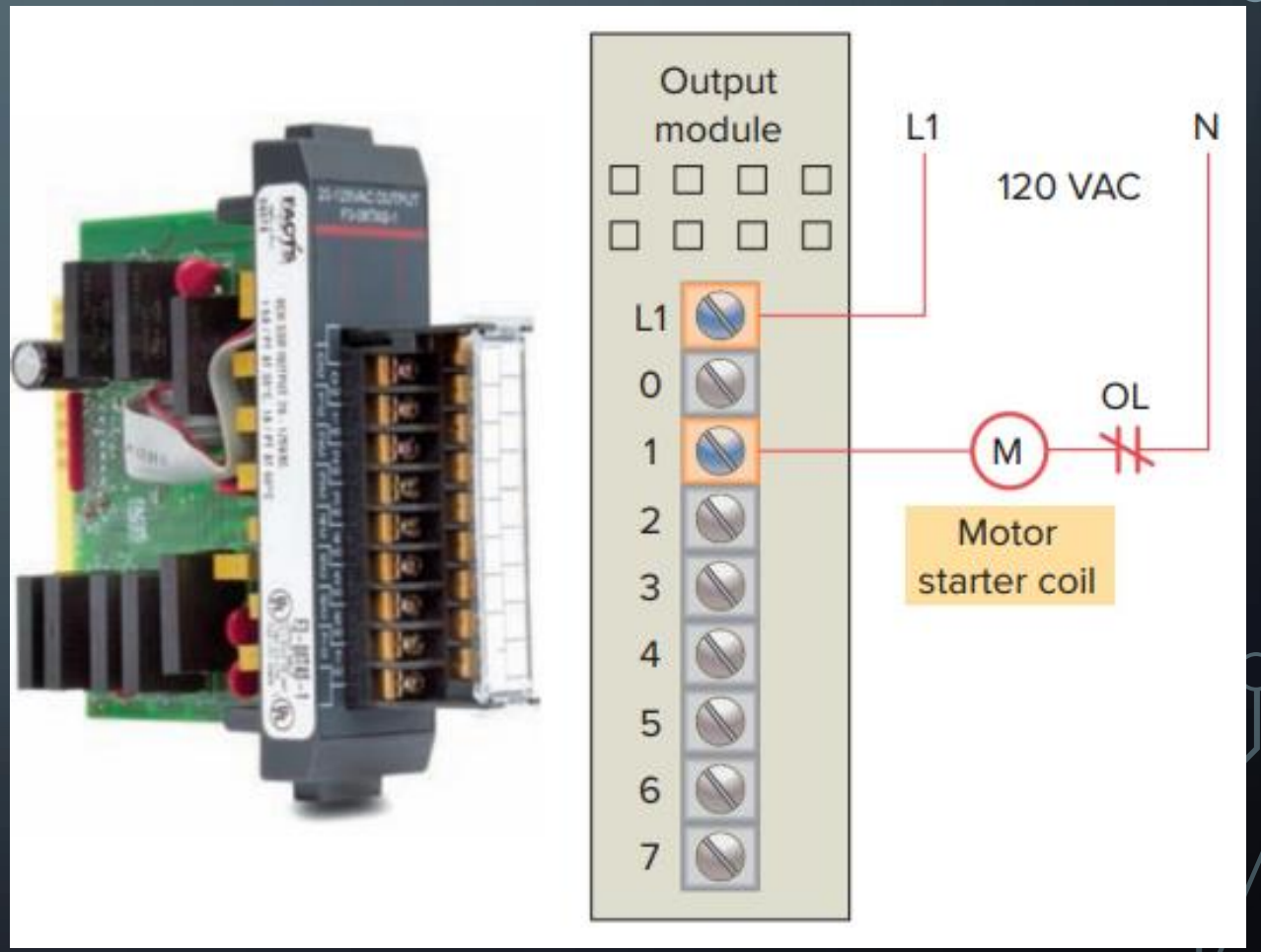
- Mixer process control: automatically stir the liquid in a vat when the temperature and pressure reach preset values



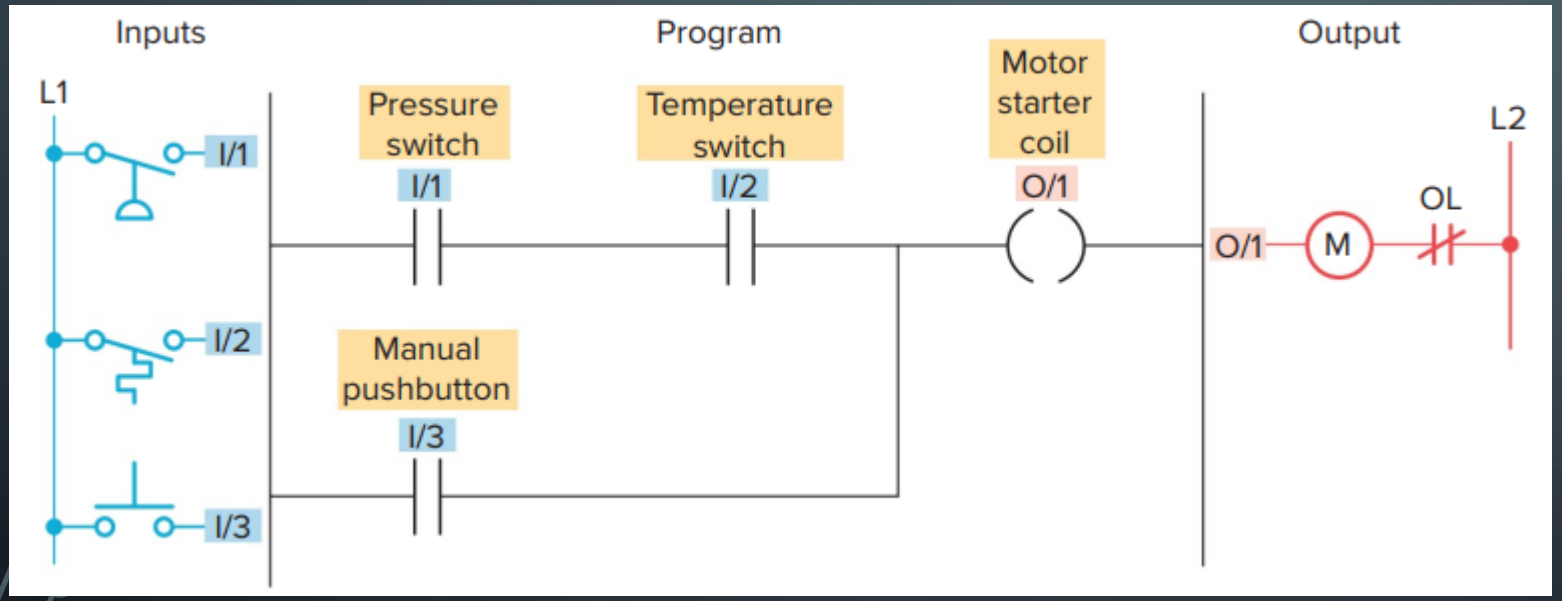


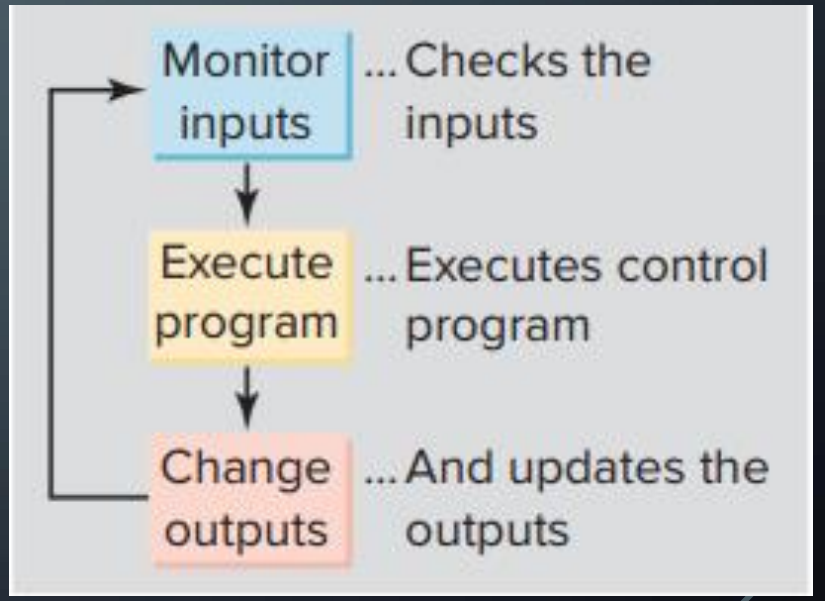
AN EXAMPLE


Input module
Pressure
Temperature
Manual pushbutton
0
1
2
3
4
5
6
7
L1
120 VAC
N
Common


Output module
L1
0
1
2
3
4
5
6
7
L1
120 VAC
N
OL
M
Motor starter coil

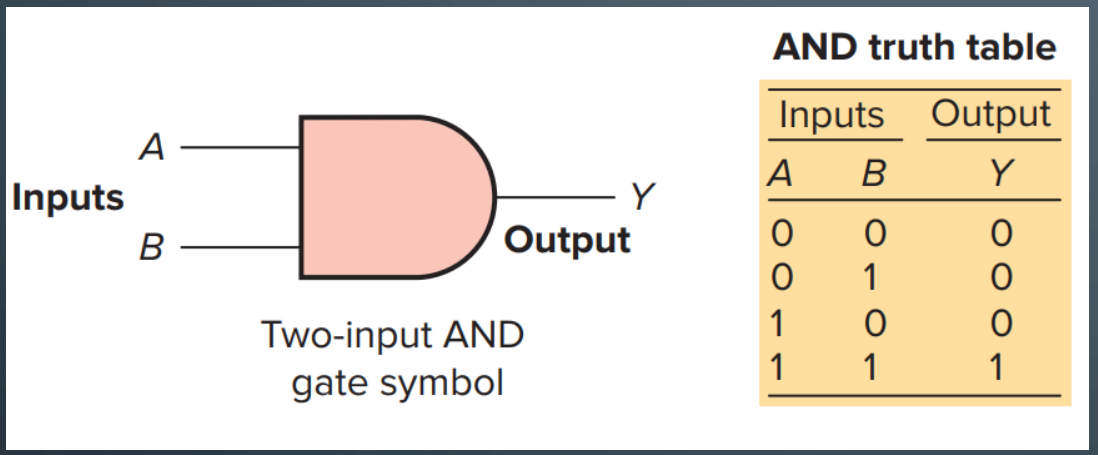
AN EXAMPLE


Inputs
Program
Output
L1
I/1
I/2
I/3
Pressure switch
I/1
Temperature switch
I/2
Manual pushbutton
I/3
Motor starter coil
O/1
O/1
M
OL
L2


Monitor inputs
... Checks the inputs
Execute program
... Executes control program
Change outputs
... And updates the outputs

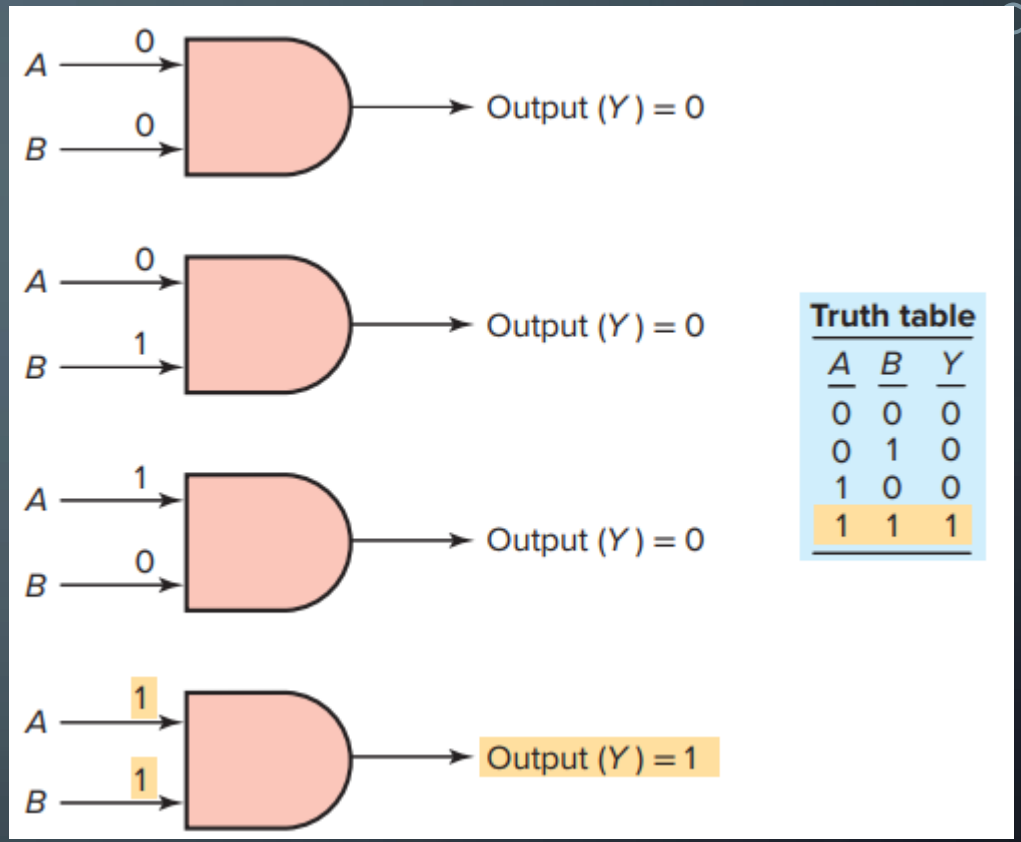
FUNDAMENTALS OF LOGIC

- And Function



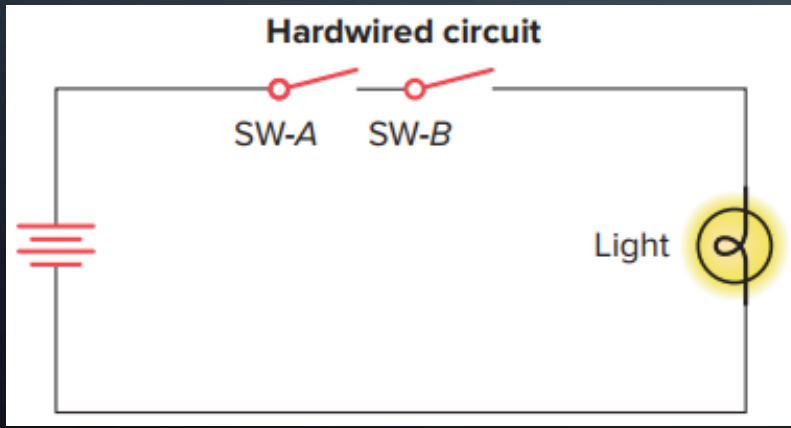
AND truth table

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



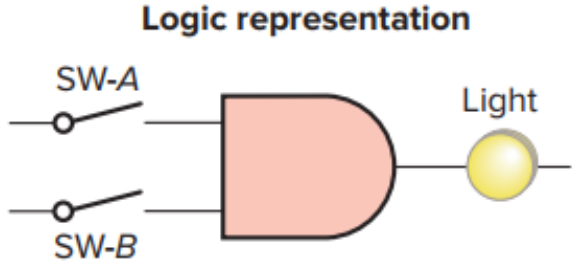
Truth table

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



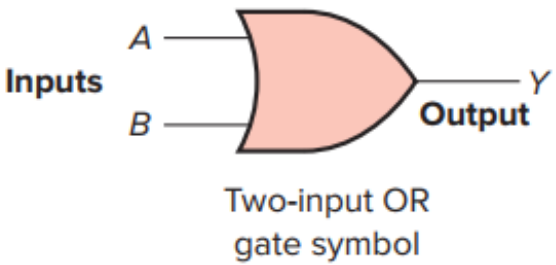
Truth table

SW- A		SW- B		Light	
Open	(0)	Open	(0)	Off	(0)
Open	(0)	Closed	(1)	Off	(0)
Closed	(1)	Open	(0)	Off	(0)
Closed	(1)	Closed	(1)	On	(1)



FUNDAMENTALS OF LOGIC

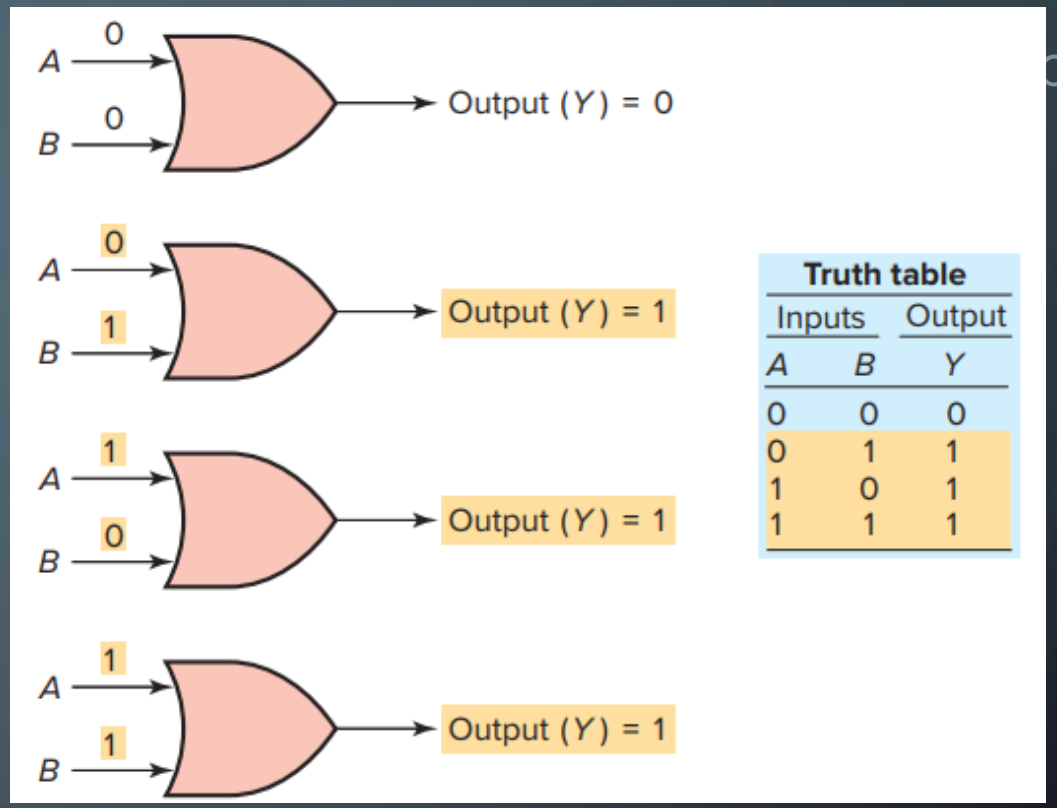
- Or Function



Two-input OR gate symbol

OR truth table

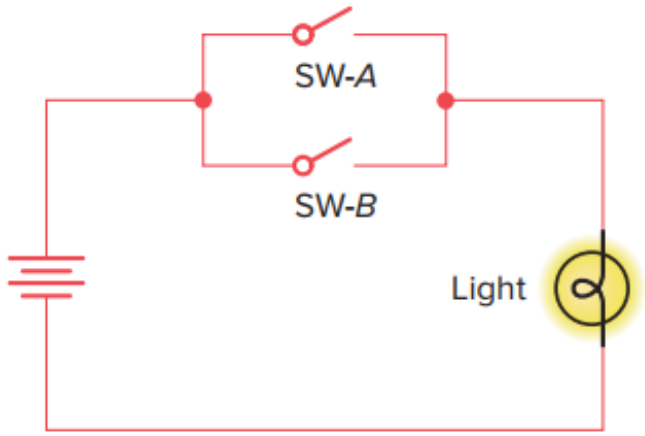
Inputs		Output
A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	1



Truth table

Inputs		Output
A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	1

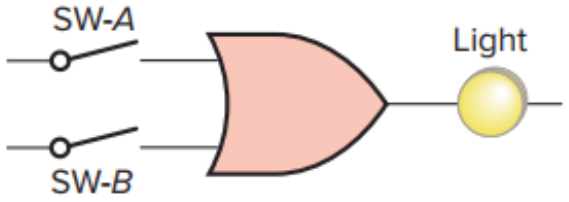
Hardwired circuit



Truth table

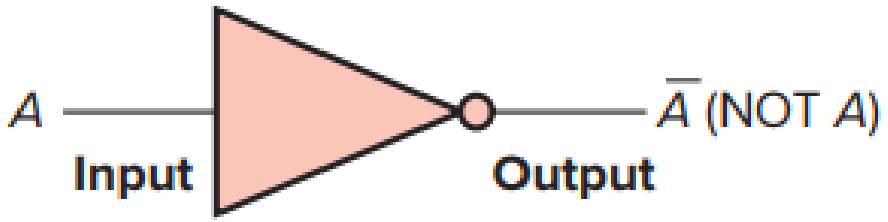
SW- *A*		SW- *B*		Light	
Open	(0)	Open	(0)	Off	(0)
Open	(0)	Closed	(1)	On	(1)
Closed	(1)	Open	(0)	On	(1)
Closed	(1)	Closed	(1)	On	(1)

Logic representation



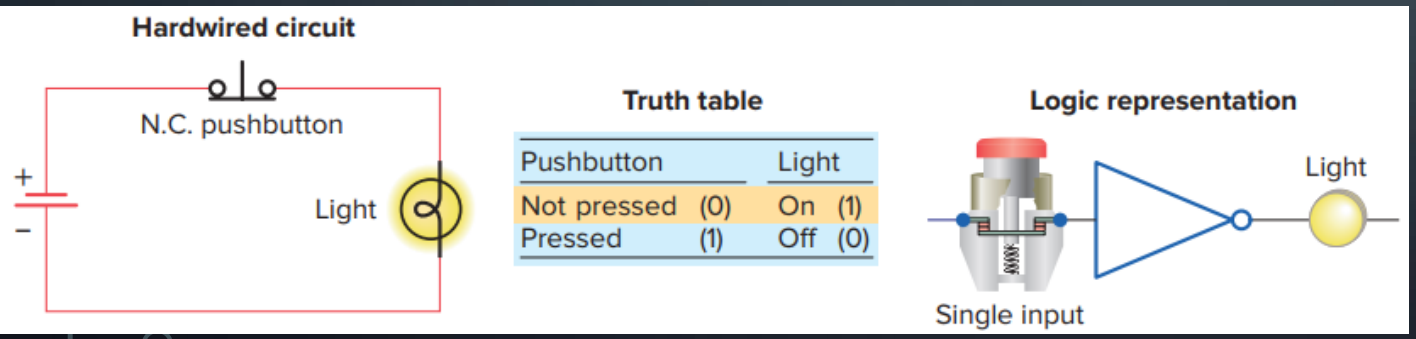
FUNDAMENTALS OF LOGIC

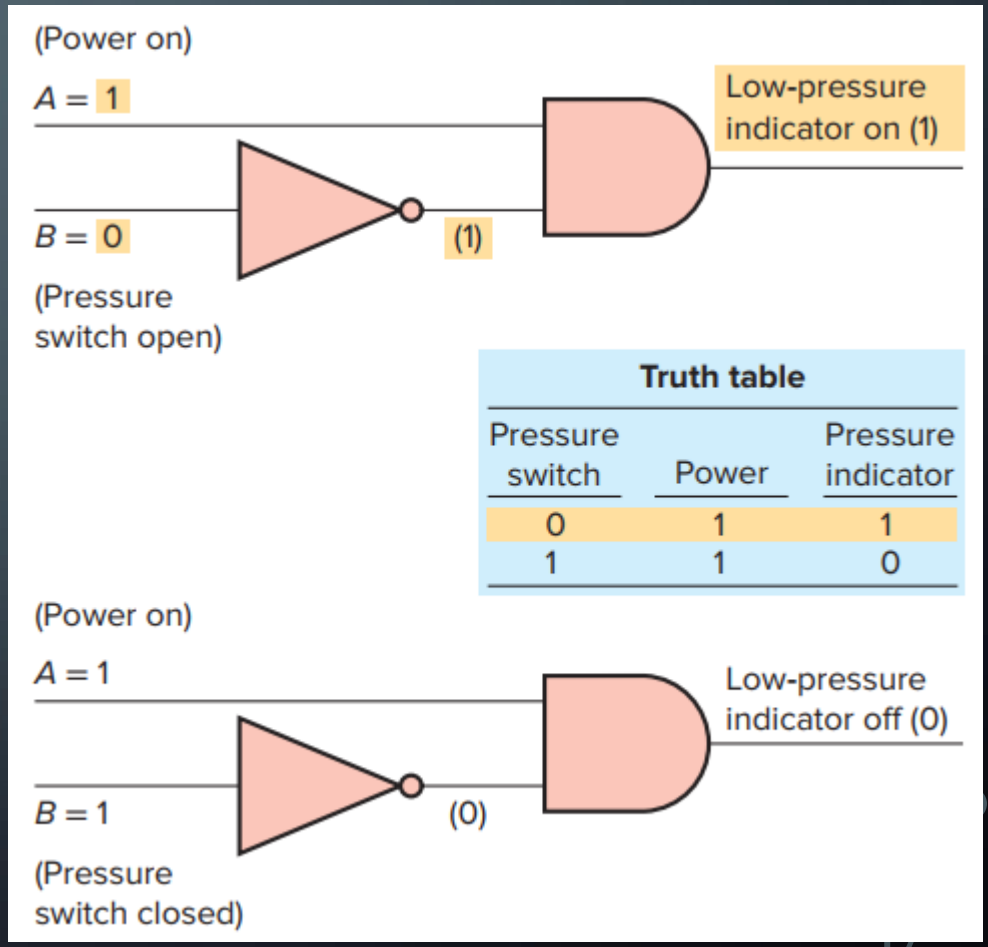
- Not Function



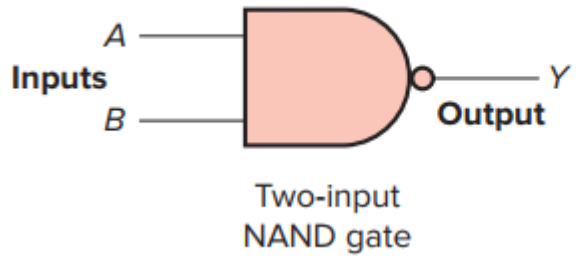
NOT truth table

A	NOT A
0	1
1	0





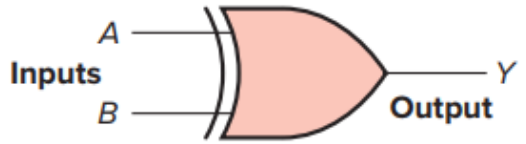
FUNDAMENTALS OF LOGIC



Two-input
NAND gate

NAND truth table

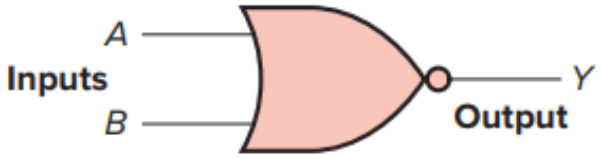
Inputs		Output
A	*B*	*Y*
0	0	1
0	1	1
1	0	1
1	1	0



Truth table

Inputs		Output
A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-OR

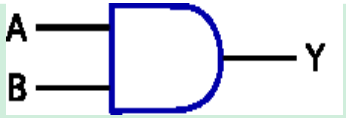
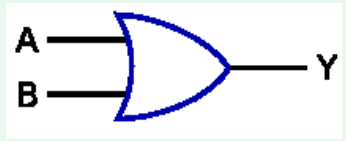
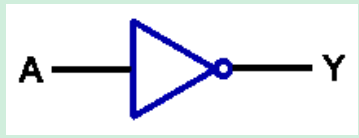
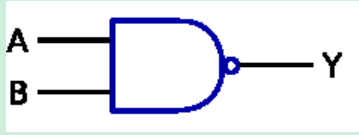
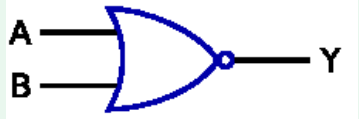


Two-input
NOR gate

NOR truth table

Inputs		Output
A	*B*	*Y*
0	0	1
0	1	0
1	0	0
1	1	0

BOOLEAN ALGEBRA

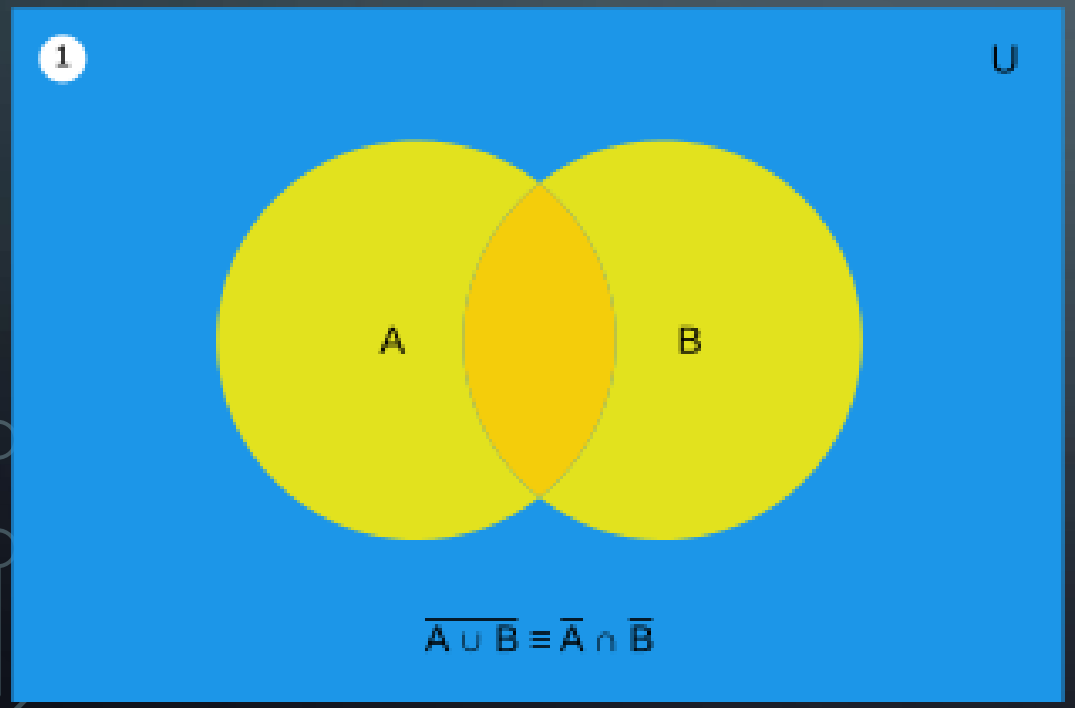
	Logic Gate	Set Theory	Binary Operations
Conjunction AND		Intersection, ∩ $A \cap B = Y$	$A \cdot B = Y$
Disjunction OR		Union, U $A \cup B = Y$	$A + B = Y$
Negation NOT		Complement - $\overline{A} = Y$	$\sim A = Y$
Exclusive disjunction XOR		$(\overline{A} \cap B) \cup (A \cap \overline{B}) = Y$	$A \oplus B = Y$
Non-conjunction NAND		$\sim(A \cap B) = Y$	$\sim(A \cdot B) = Y$
Non-disjunction NOR		$\sim(A \cup B) = Y$	$\sim(A + B) = Y$

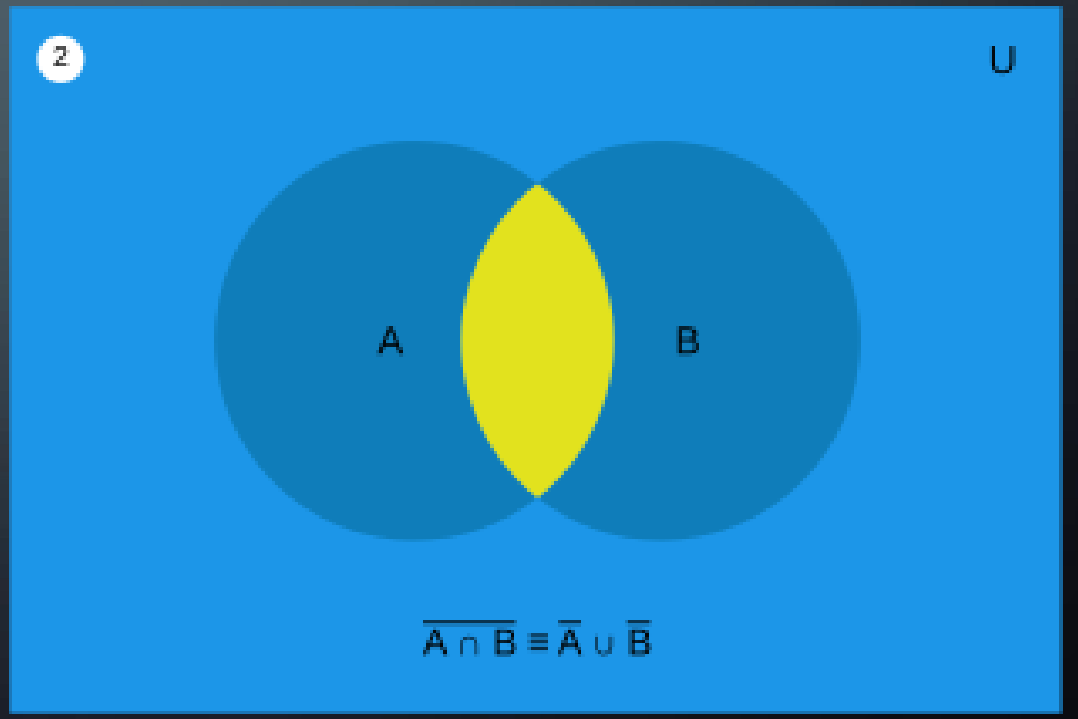
PROPERTIES

Associative property	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A+B)+C = A+(B+C)$
Commutive property	$A \cdot B = B \cdot A$	$A + B = B + A$
Distributive property	$A \cdot (B+C) = A \cdot B + A \cdot C$	$A+(B \cdot C) = (A+B) \cdot (A+C)$
Identity	$A \cdot 1 = A$	$A + 0 = A$
Inverse	$A \cdot \sim A = 0$	$A+\sim A = 1$
Absorption	$A \cdot (A+B) = A$	$A+(A \cdot B) = A$
deMorgan's theorem	$\sim(A \cdot B) = \sim A + \sim B$	$\sim(A+B) = (\sim A) \cdot (\sim B)$

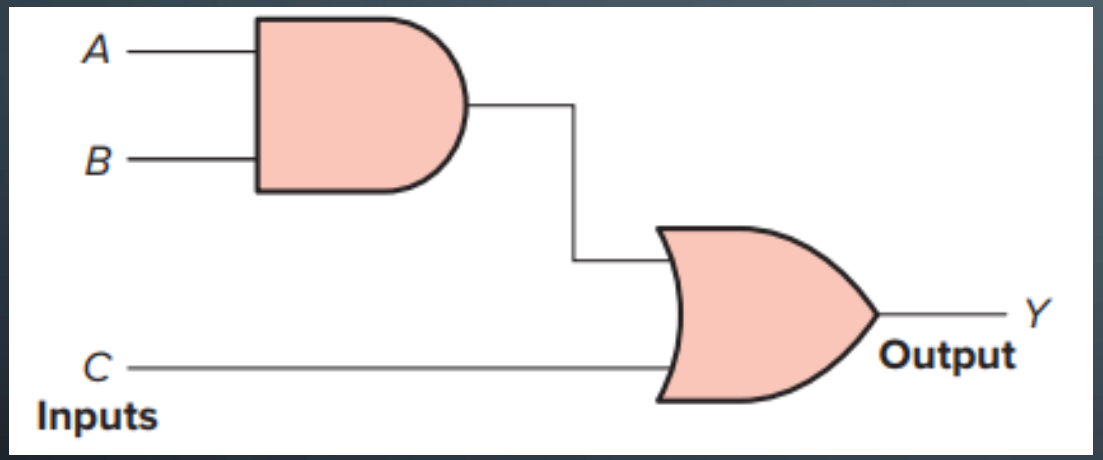
DE MORGAN'S LAW

- the negation of a disjunction is the conjunction of the negations
- the negation of a conjunction is the disjunction of the negations

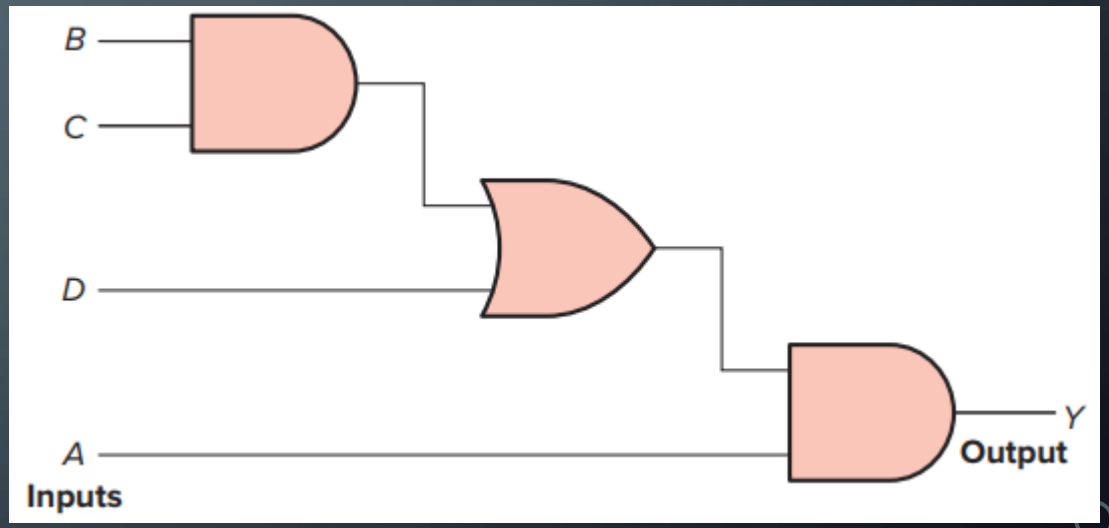




LOGIC GATE CIRCUITS & BOOLEAN EQUATION

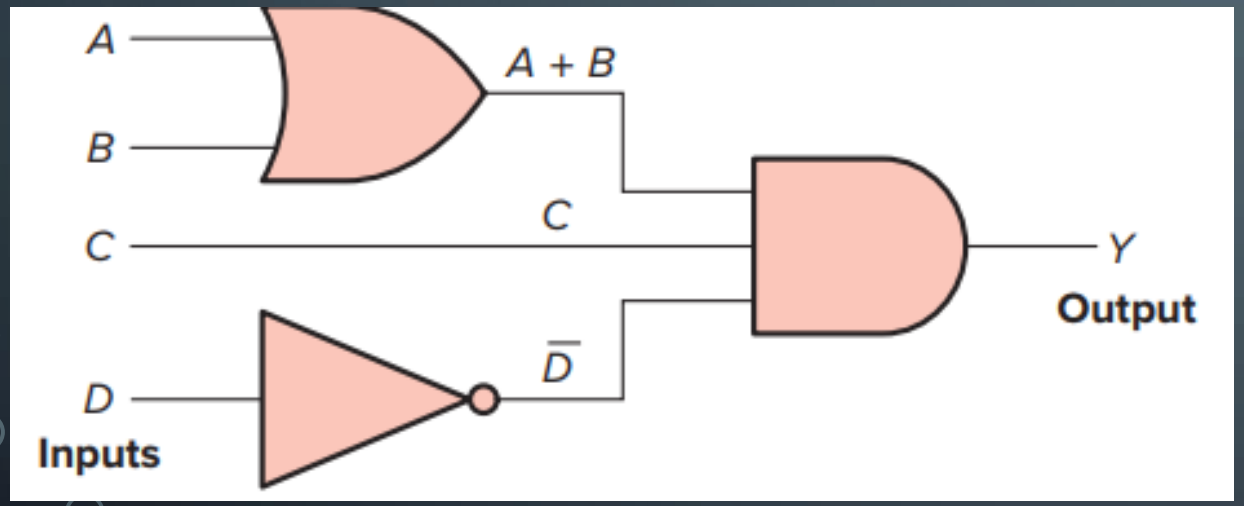


Boolean expression: Y = AB + C

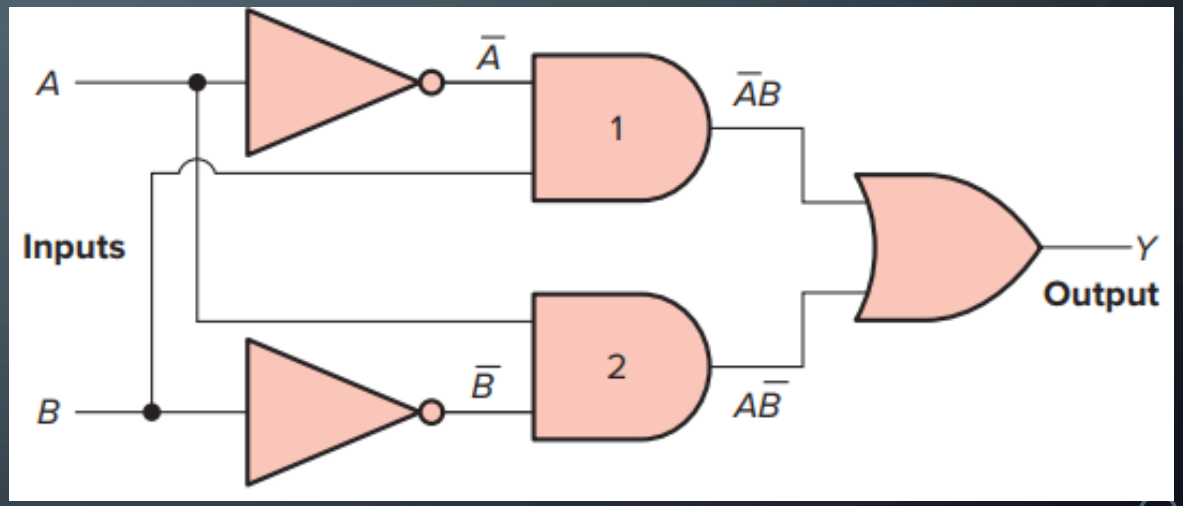


Boolean expression: Y = A(BC + D)

LOGIC GATE CIRCUITS & BOOLEAN EQUATION

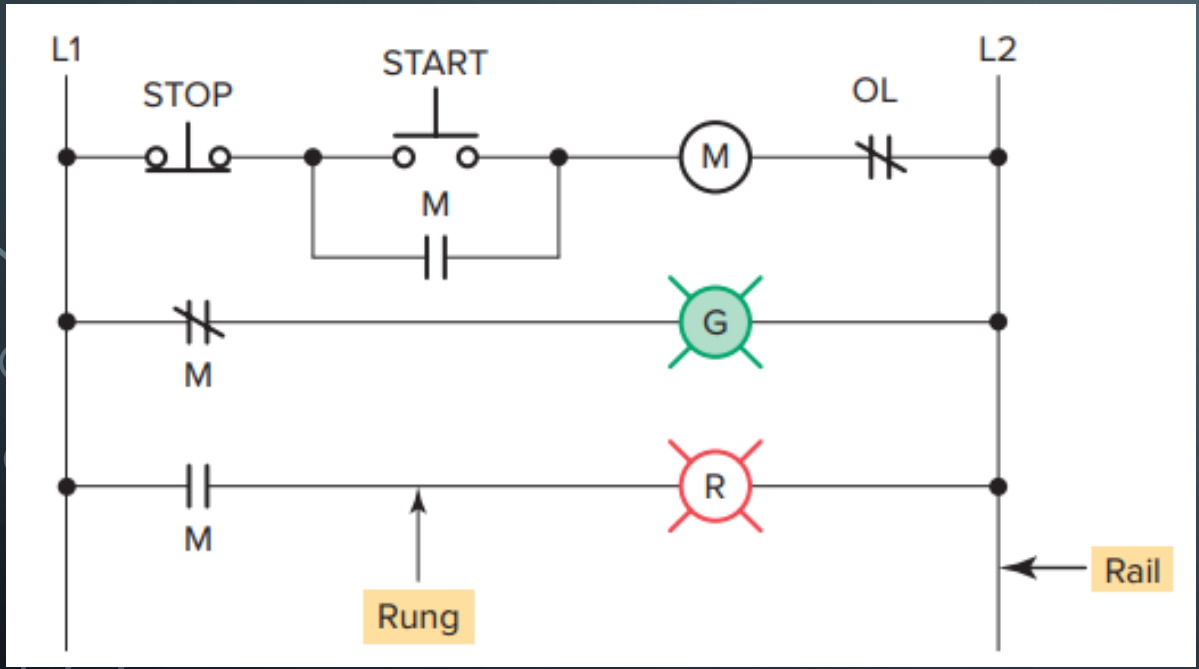


Boolean expression: $Y = C\overline{D}(A + B)$

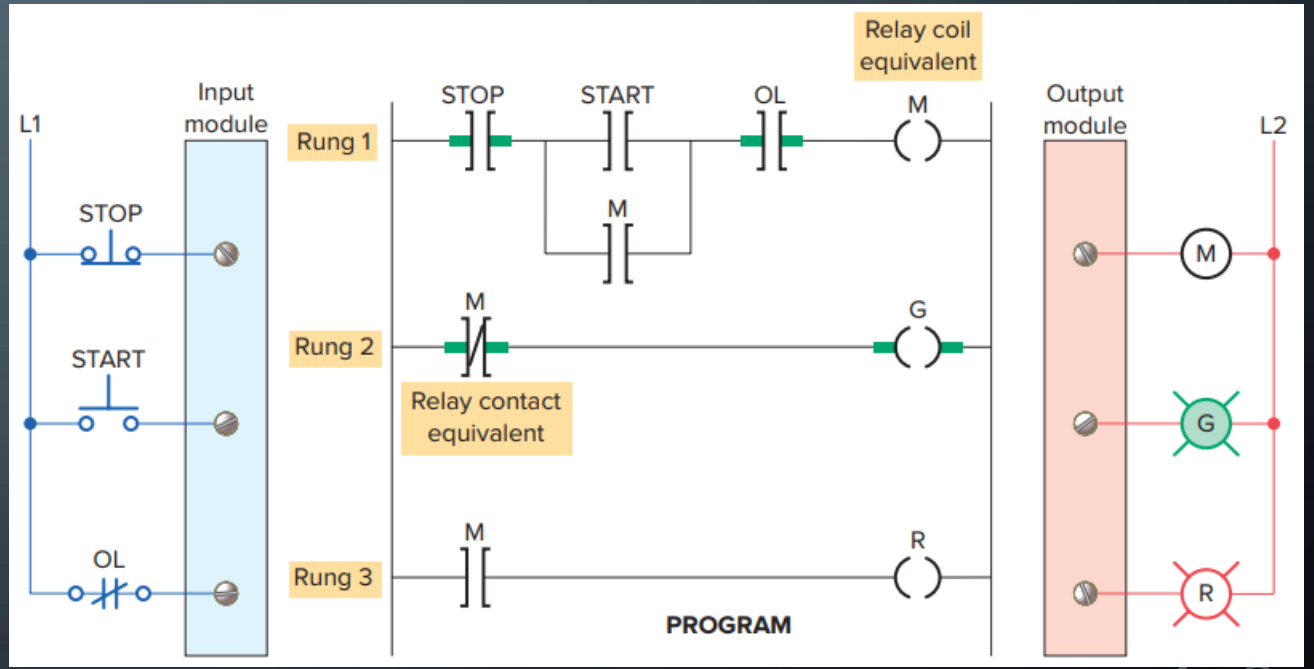


Boolean expression: $Y = \overline{A}B + A\overline{B}$

HARDWIRED LOGIC VS PROGRAMMED LOGIC

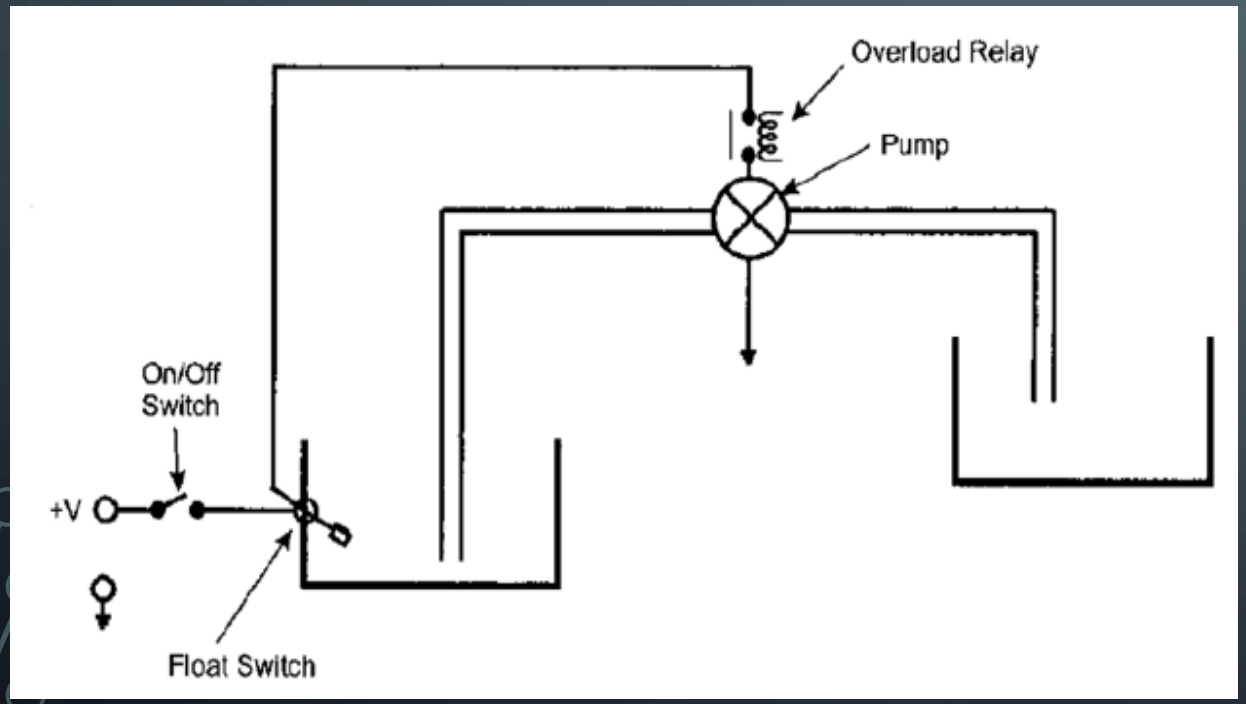


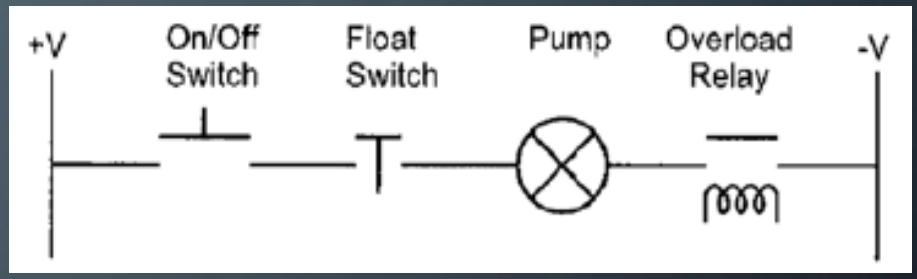
Motor stop/start relay ladder schematic

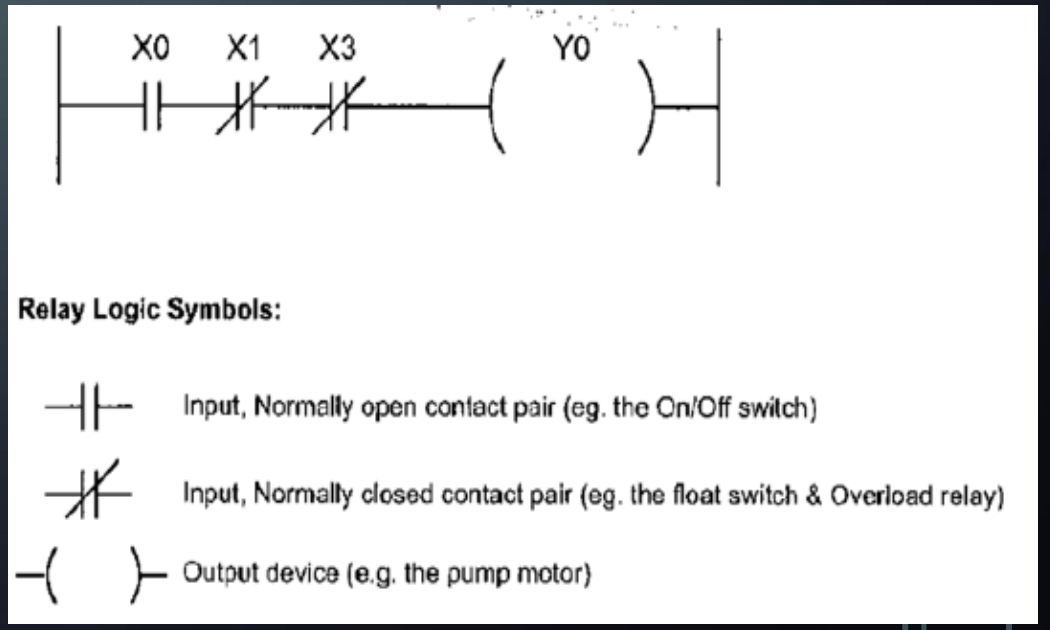


Motor stop/start ladder logic program

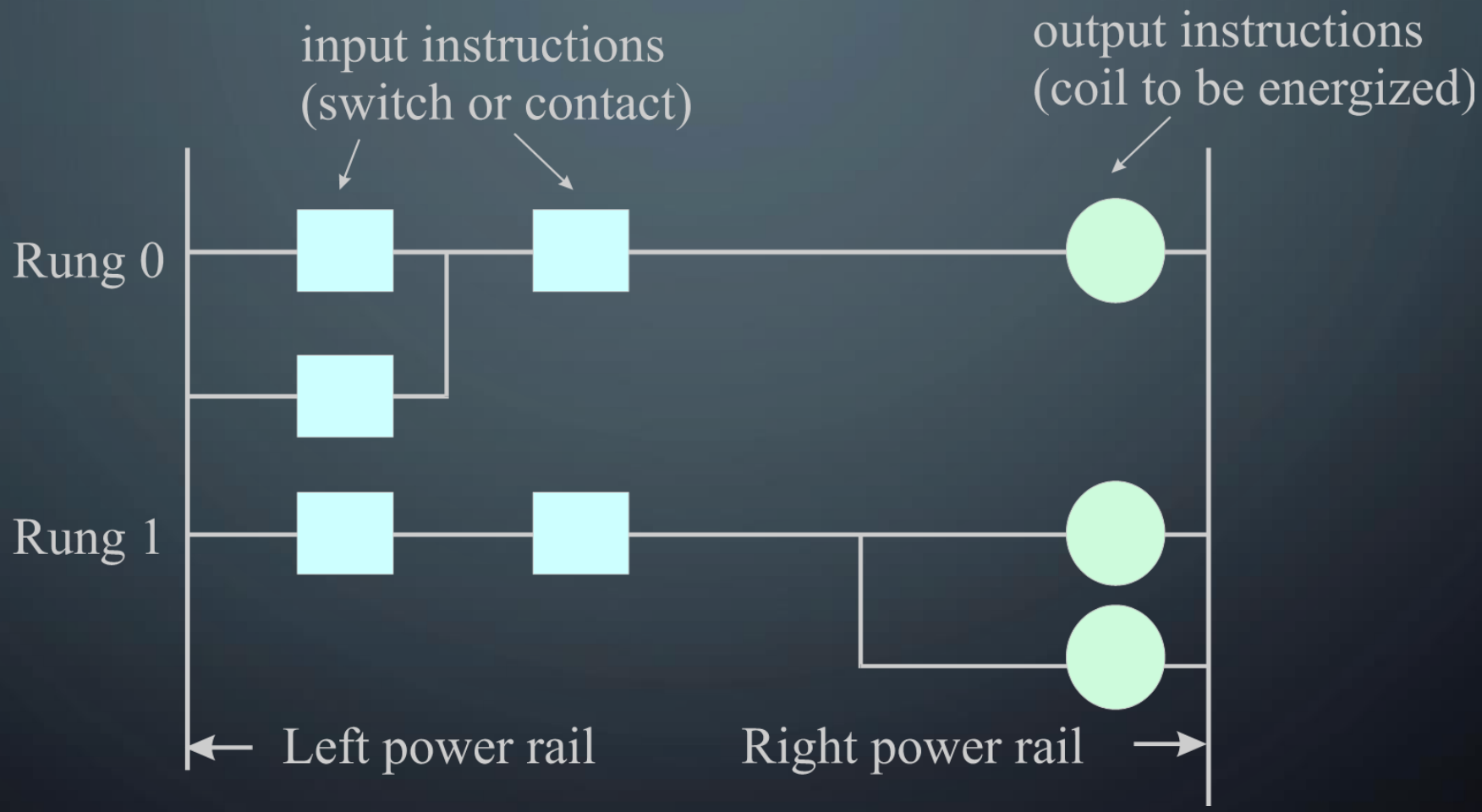
HARDWIRED LOGIC VS PROGRAMMED LOGIC







LADDER LOGIC



LADDER LOGIC

- Ladder logic strongly resembles a schematic diagram of relay logic.
- Ladder logic notation is chosen to reduce training demands for the existing technicians.
- Input instructions are entered on the left (left power rail).
- Output instructions are entered on the right.
- The output will execute only if the connection is made from left to right power rail.
- The program is executed from top to bottom, left to right.

LADDER LOGIC

- There are both ***contacts*** and ***coils*** that can be loaded and driven in different configurations that make up a ladder diagram.
- Devices are actuated by or act upon variables associated with the PLC hardware.
- All the variables are logical, and take values either OFF (Logical '0') or ON (Logical '1').

Instruction	Symbol
Contact (Normally Open)	
Contact (Normally Closed)	
Coil	

X	This is used to represent the physical inputs to the PLC. There are 8 physical inputs to the CE123 addressed as **X0 — X7**
Y	This represents the physical outputs of the PLC. There are 6 physical outputs to the CE123 addressed as **Y0 – Y5**
T	This represents a timing device in the PLC. There are 64 timers in the CE123 addressed as T0 to T63. The CE123 timers are set in units of 0.1 seconds but can be reprogrammed. Timers and counters will be further discussed below
C	This represents a counting device in the PLC. There are 32 counters in the CE123 addressed as C0-C31
M	These are used to represent internal operational flags in the PLC. There are 512 M flags.

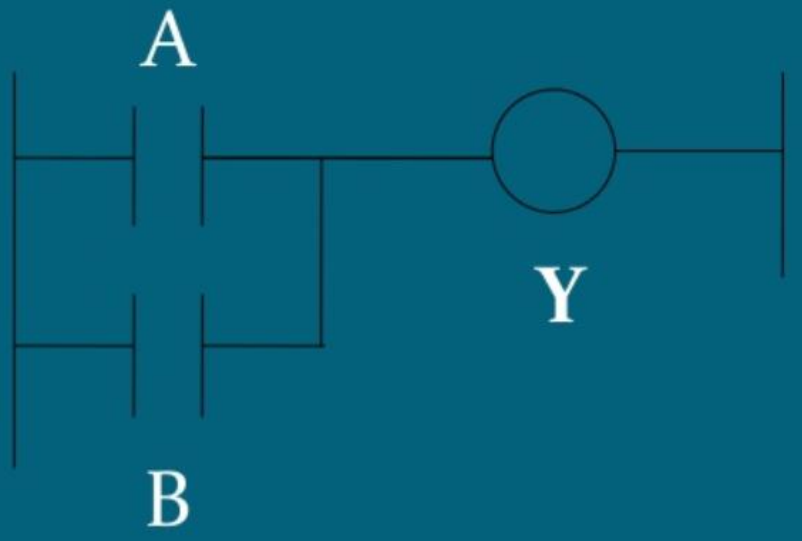
Ladder Logic For Basic gates

AND Gate

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

OR Gate

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

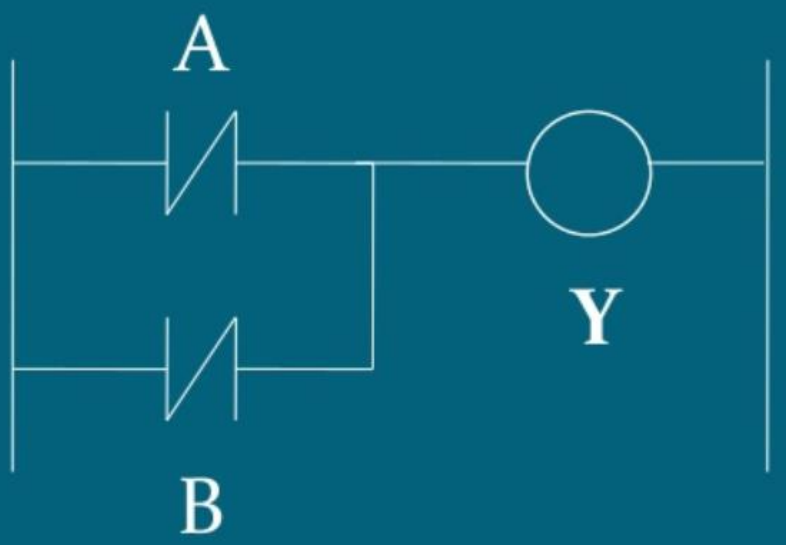


NOR Gate

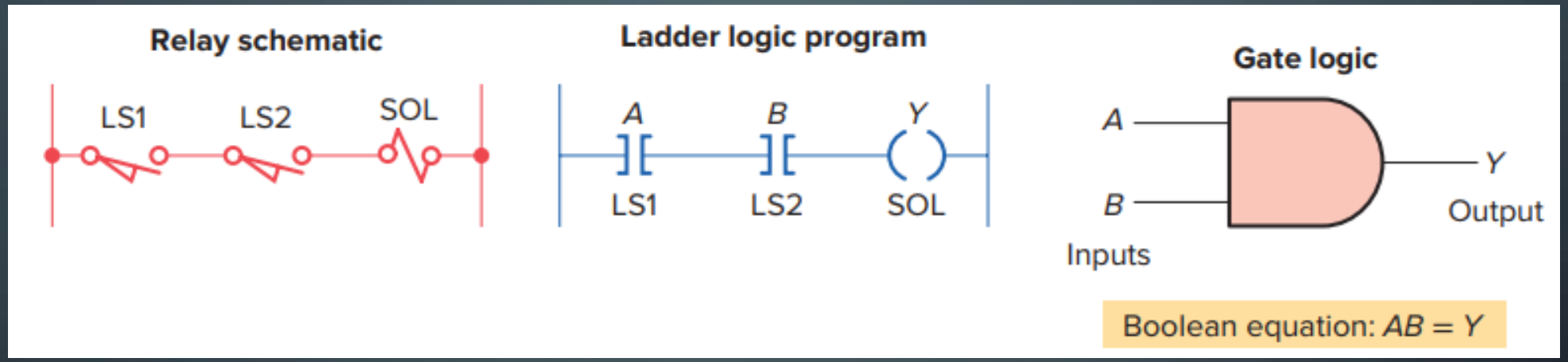
A	B	Logic(Y)
OFF	OFF	ON
OFF	ON	OFF
ON	OFF	OFF
ON	ON	OFF

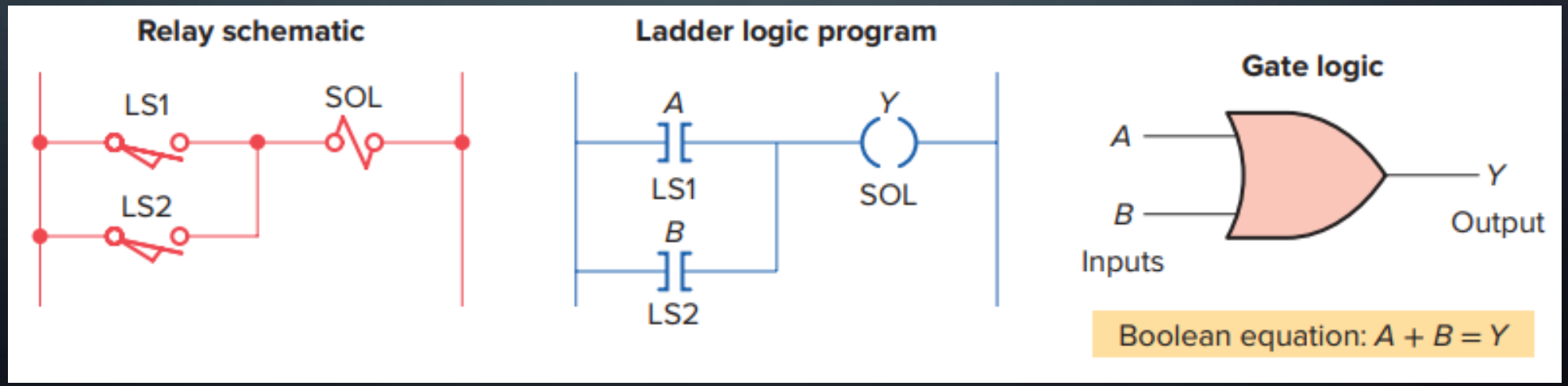
NAND Gate

A	B	Logic(Y)
OFF	OFF	ON
OFF	ON	ON
ON	OFF	ON
ON	ON	OFF

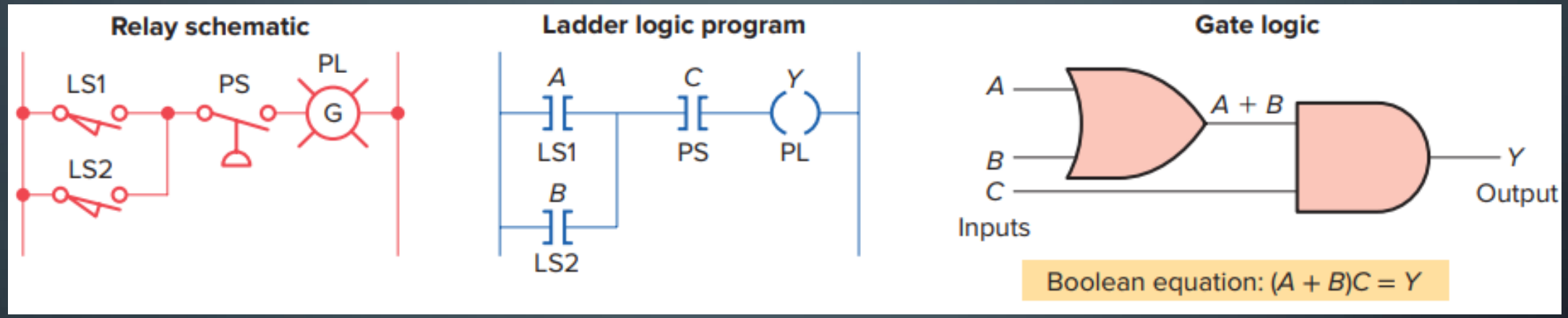


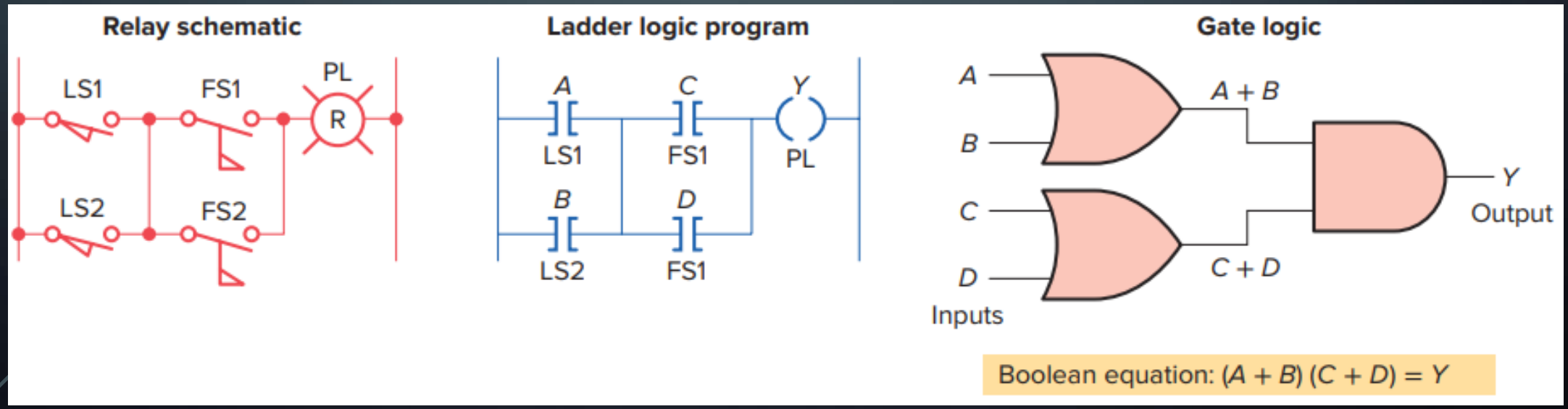
SAMPLE LADDER LOGIC PROGRAM





SAMPLE LADDER LOGIC PROGRAM



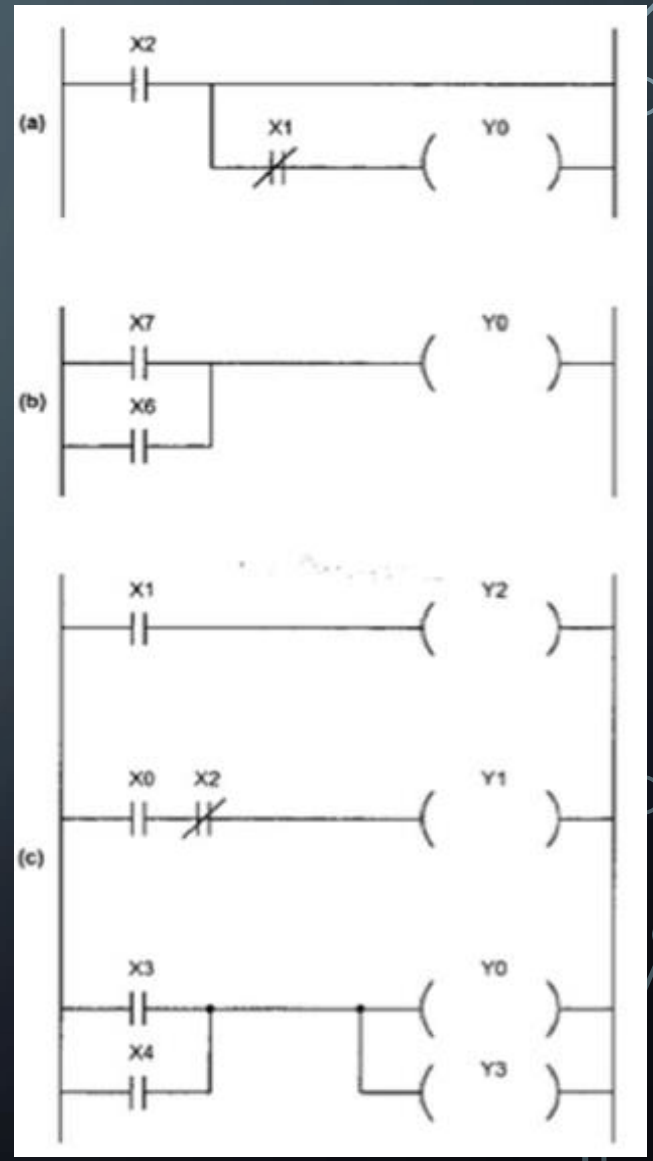


LADDER LOGIC EXAMPLE

- (a) Represents a system comprised of a motor, Y0, which will only operate if switch X2 (normally open) has been closed and switch X1 (normally closed) has not been triggered.

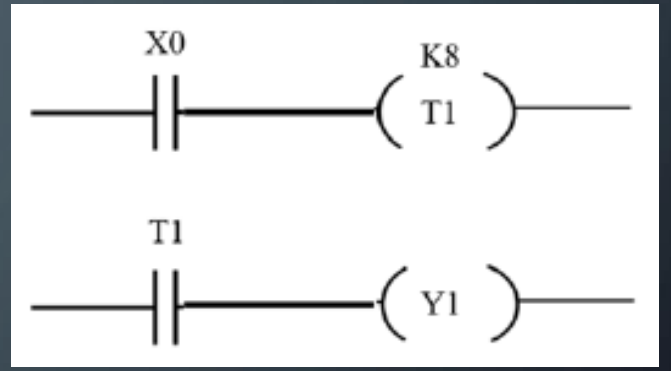
- (b) Represents a system comprised of an output device, Y0, which will operate if either of two normally open switches, X6 or X7, are closed.

- (c) Shows a more complex diagram: output Y2 will be turned on if the contact, X1, is closed. Output Y1 is controlled by the state of witches X0 and X2, only being activated if X0 is closed and X2 is not opened. Outputs Y0 and Y3 will both be activated if either switch, X3 or X4 are closed.



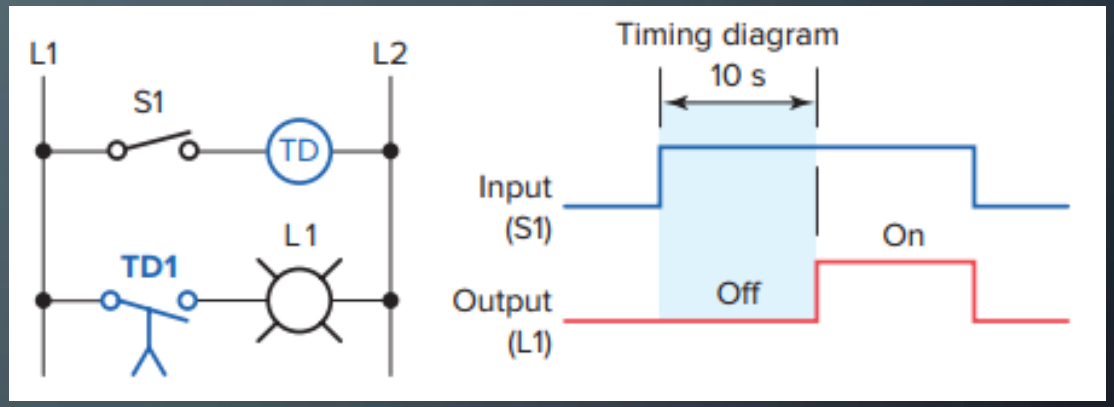
TIMERS AND FLAGS

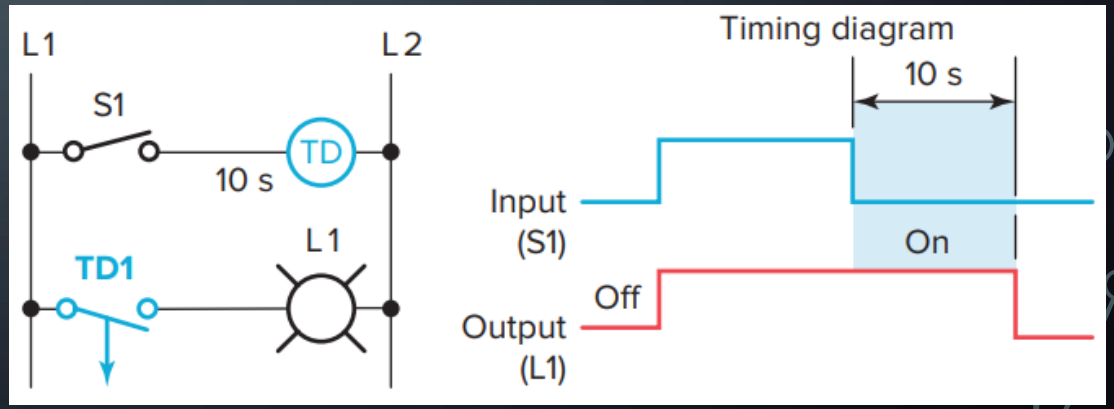
- Timers and Flags can be used to drive an output (coil) or they can be used to trigger an input (contact).
- When configuring a coil (output) for use as a timer, a constant must also be entered. This constant, identified by the letter “K”, tells the program how long that timer will run.
- If you enter K8 for a timer constant, the duration data will be held for 800 ms (8 x 100ms) or 0.8 seconds.
- When input X0 is opened, T1 is initiated and holds for 0.8 seconds before executing (opening). After 0.8 seconds, output Y1 is triggered.
- The number next to the M is used only as identification. Instead of writing the same code (which may be many lines long) you can store that line as a flag and call it when needed.



TIMERS

- ON delay timer
  - When the input to an ON DELAY goes from false to true, the output goes to true after delaying a preset time interval (called TON)
  - When the input to an ON DELAY goes from true to false, the output immediately goes to false.
- OFF delay timer
  - When the input to an OFF DELAY goes from true to false, the output goes to false after delaying a preset time interval (called TOFF).
  - When the input to an OFF DELAY goes from false to true, the output immediately goes to true.





LAB 3 PLC

- Goal: To investigate the fundamentals of programmable logic controllers (PLCs) and their operation as it relates to fluid transport and liquid level
- Examine the time it takes to drain tank 1 to a specific level

TIMER CONSTANT CALCULATION

- Torricelli's theorem

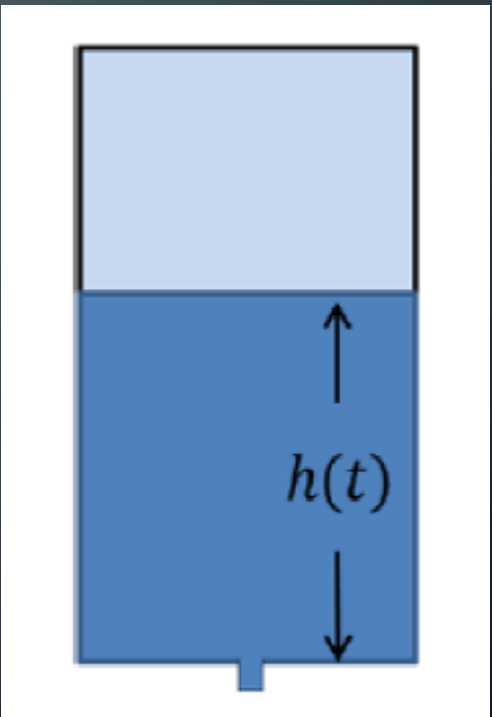
$$V = \sqrt{2gh}$$

- Conservation of mass

$$\frac{\delta m_{cv}}{\delta t} + \dot{m}_{out} = 0$$

- Exit mass flow rate

$$\dot{m}_{out} = \rho V A_o$$



TIMER CONSTANT CALCULATION

- Rewrite the equation

$$m_{cv} = \rho A_t h$$

$$\rho A_t \frac{dh}{dt} + \rho A_o \sqrt{2gh} = 0$$

- Separate variables and integrate

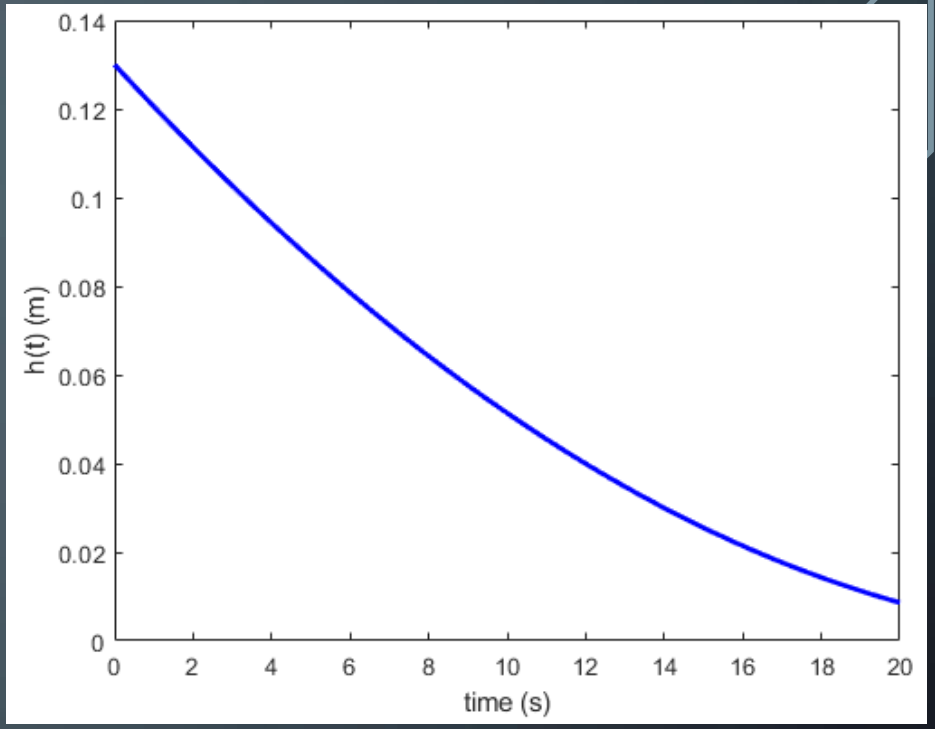
$$A_t \frac{dh}{\sqrt{h}} = -A_o \sqrt{2g} dt$$

$$2A_t \left(\sqrt{h(t)} - \sqrt{h(0)}\right) = -A_o \sqrt{2g} t$$

$$\sqrt{h(t)} = \sqrt{h(0)} - \frac{A_o}{A_t} \sqrt{\frac{g}{2}} t$$

TIMER CONSTANT CALCULATION

- Tank Diameter: 90mm
- Outlet Diameter: 7mm
- Tank Height (Max Fluid Height): 130mm



$$\sqrt{h(t)} = \sqrt{0.13} - \frac{0.007^2}{0.09^2}\sqrt{\frac{9.8}{2}}t$$

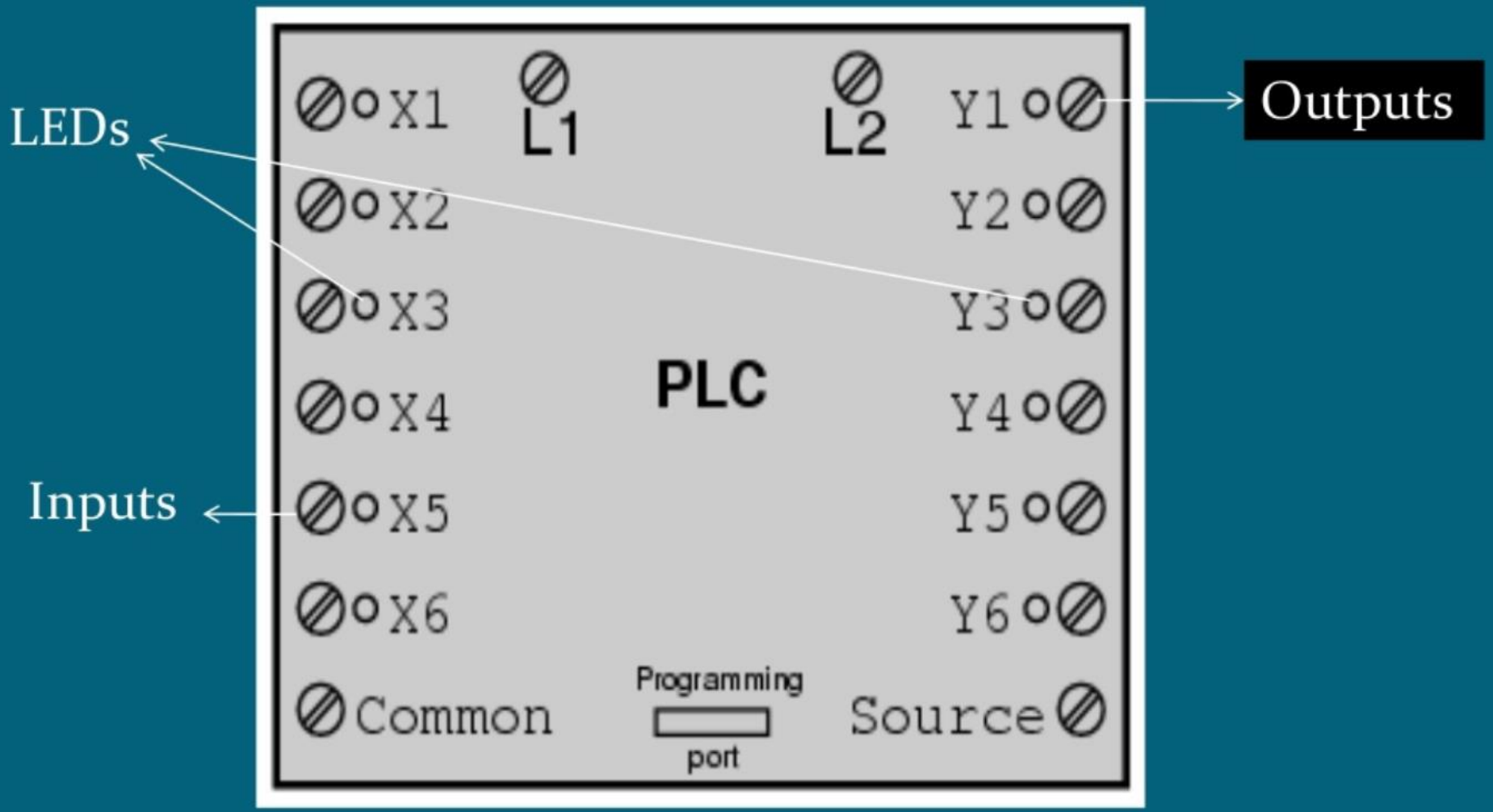
$$h(t) = (0.36 - 0.0134t)^2$$

GX DEVELOPER SOFTWARE

- The menu driven software package supplied with the CE123 allows you to:
  - Create and edit ladder diagrams
  - Download and upload ladder diagrams to and from the PLC
  - Load and save ladder diagrams to and from disk
  - Add comments to aid ladder diagram clarity
  - Monitor ladder diagrams in the PLC
  - Print ladder diagrams and comments

- Operation of the GX developer software will be investigated in the Studio Assignment.

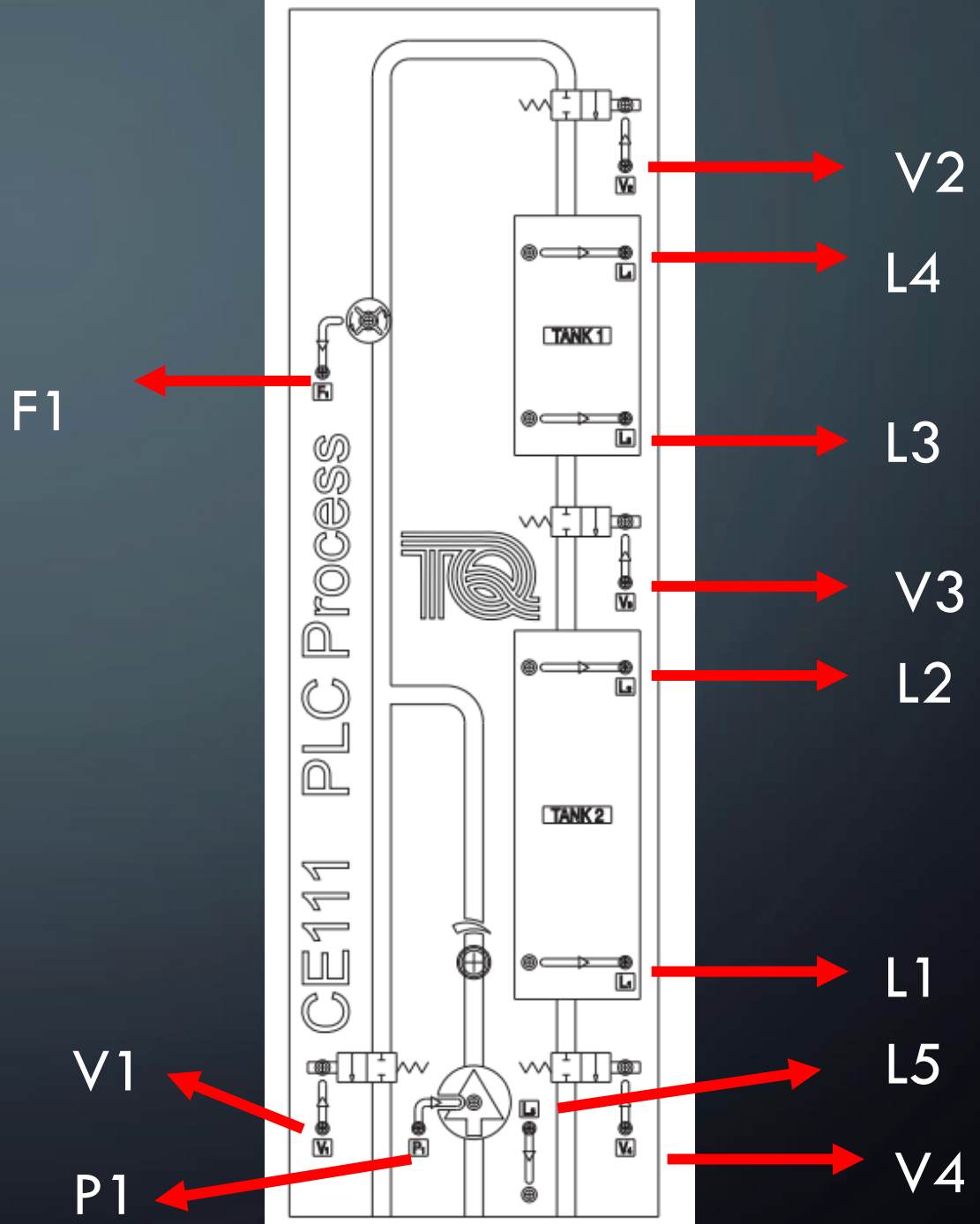
Block diagram of a PLC

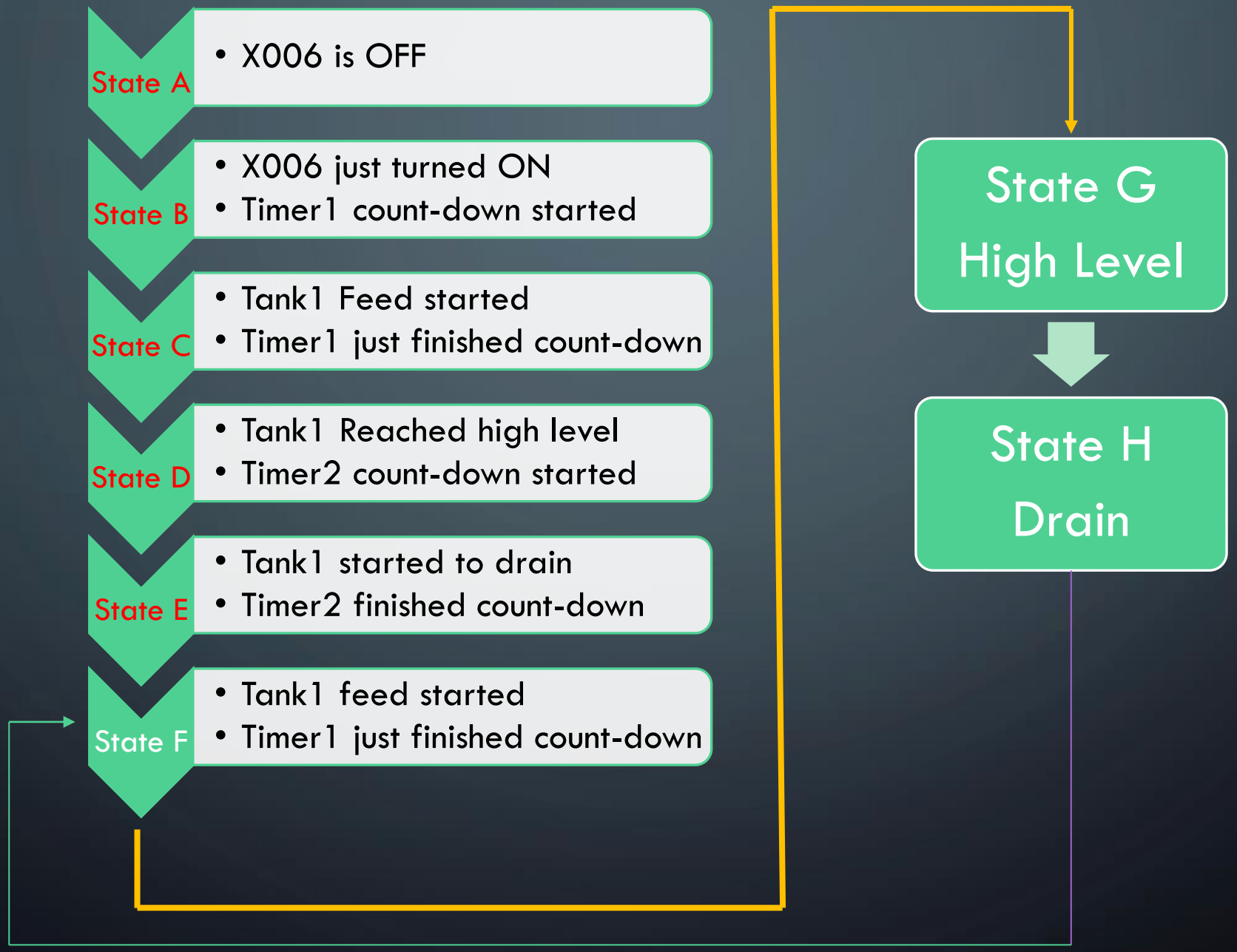


Socket Label	I/O	Function
L1	Output	Tank 2 LOW level Switch
L2	Output	Tank 2 HIGH level Switch
L3	Output	Tank 1 LOW level Switch
L4	Output	Tank 1 HIGH level Switch
L5	Output	Reservoir LOW level Float Switch
V1	Input	Pump to Reservoir Valve (by-pass)
V2	Input	Tank 1 Feed Valve
V3	Input	Tank 1 Drain Valve
V4	Input	Tank 2 Drain Valve
P1	Input	Pump ON/OFF
F1	Output	Flow meter, pulse related to flow rate

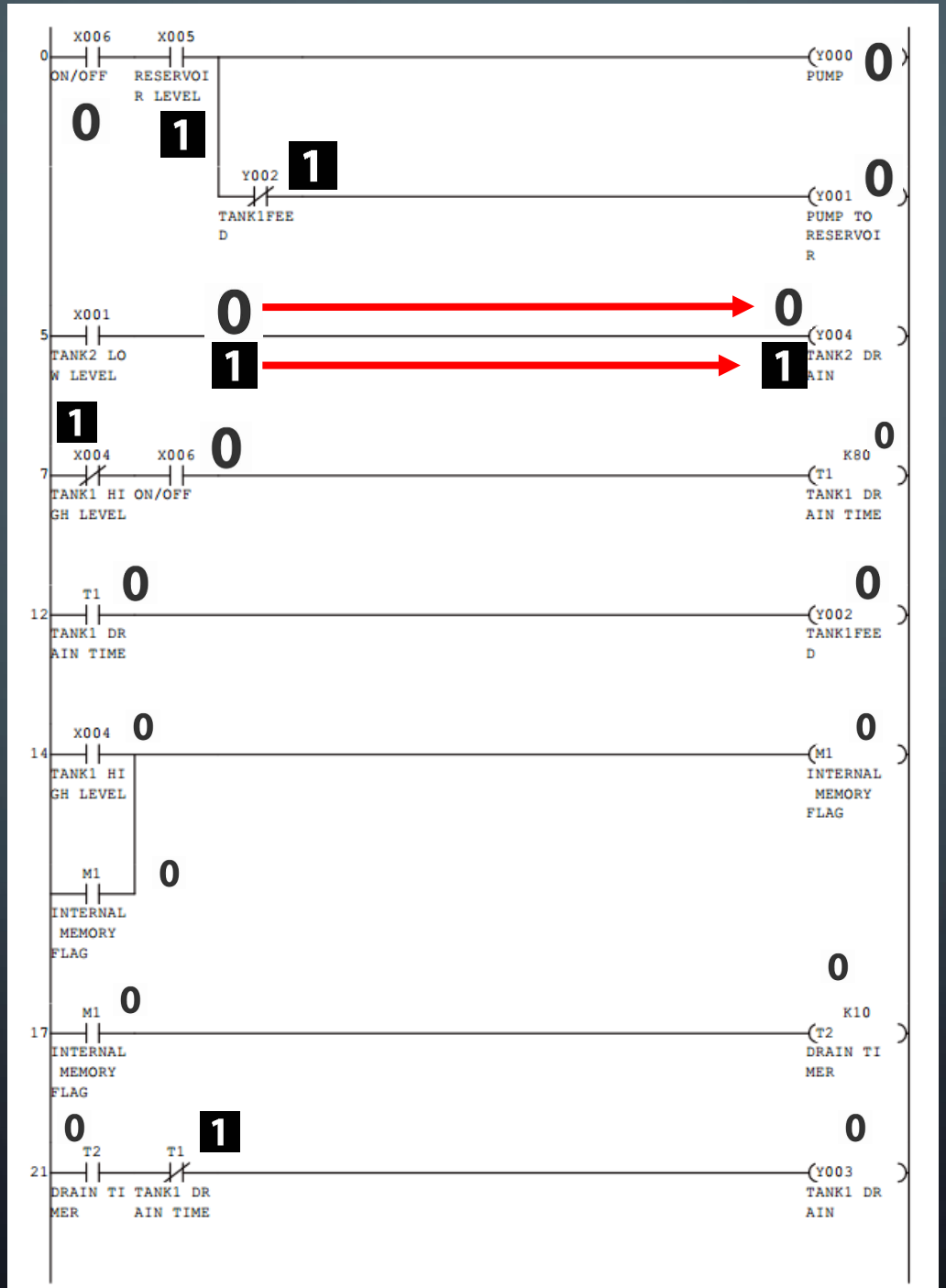
Table 2.1: Connection of Inputs/Outputs between CE111 and CE123

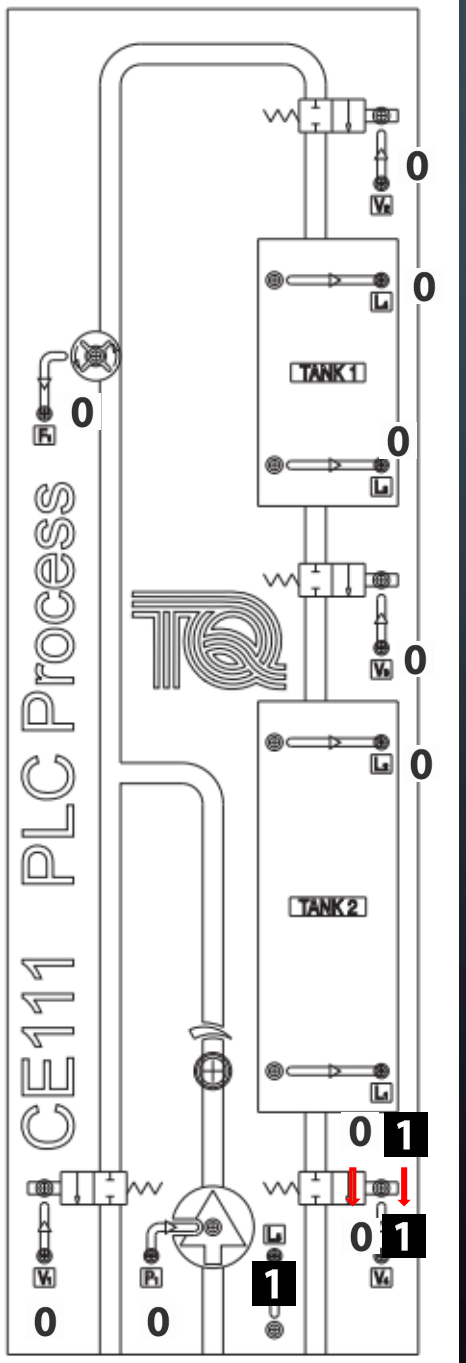
CE111	CE123	Function
L1	X1	Tank 2 LOW level Float Switch
L2	X2	Tank 2 HIGH level Float Switch
L3	X3	Tank 1 LOW level Float Switch
L4	X4	Tank 1 HIGH level Float Switch
L5	X5	Reservoir LOW level Float Switch
V1	Y1	Pump to Reservoir Valve (by-pass)
V2	Y2	Tank 1 Feed Valve
V3	Y3	Tank 1 Drain Valve
V4	Y4	Tank 2 Drain Valve
P1	Y0	Pump ON/OFF
F1	X0	Flow meter, pulse related to flow rate




State A
• X006 is OFF
State B
• X006 just turned ON
• Timer1 count-down started
State C
• Tank1 Feed started
• Timer1 just finished count-down
State D
• Tank1 Reached high level
• Timer2 count-down started
State E
• Tank1 started to drain
• Timer2 finished count-down
State F
• Tank1 feed started
• Timer1 just finished count-down
State G
High Level
State H
Drain

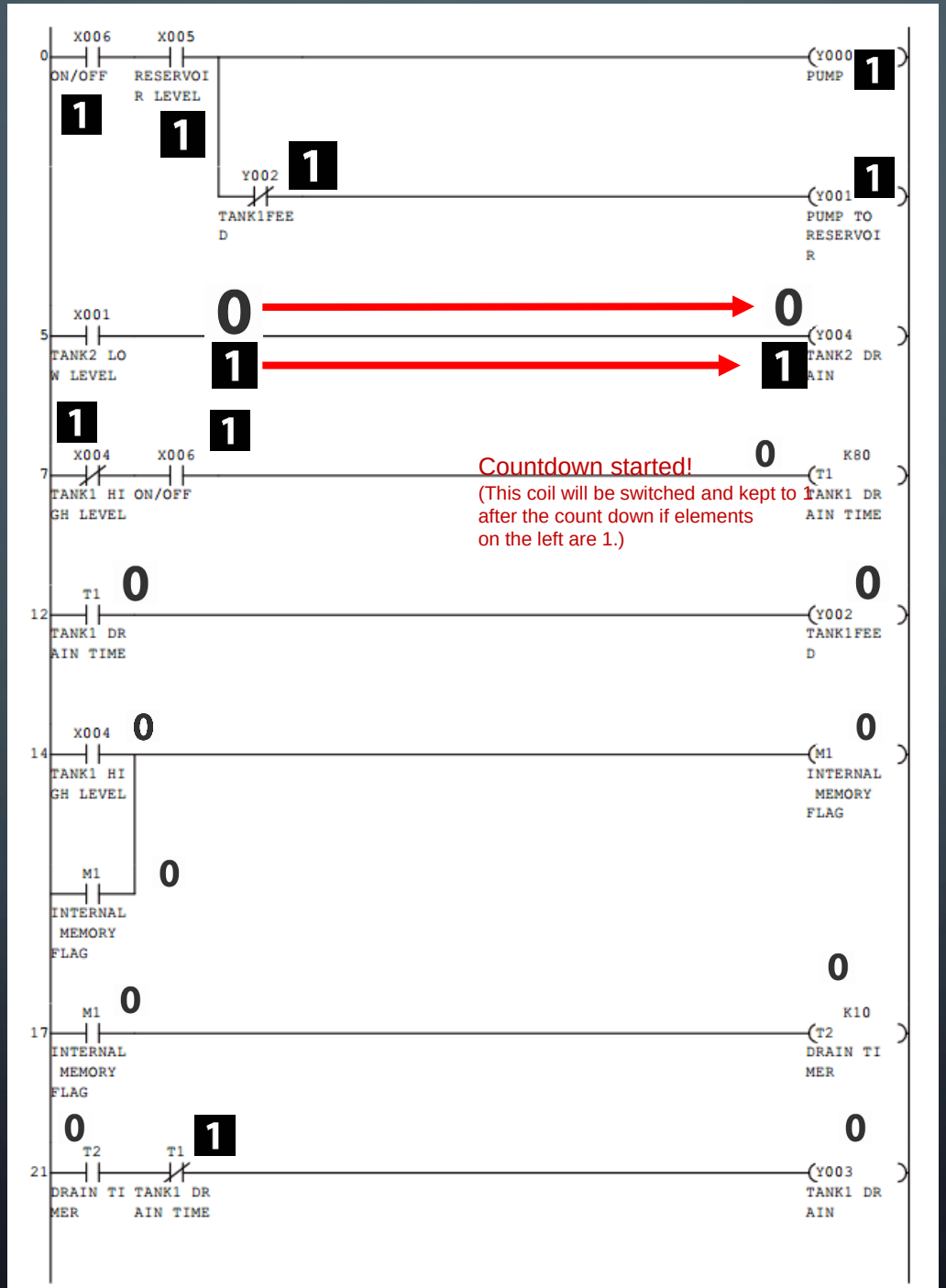
State A:
X006 was OFF
- Assume that there is no water in either tank.

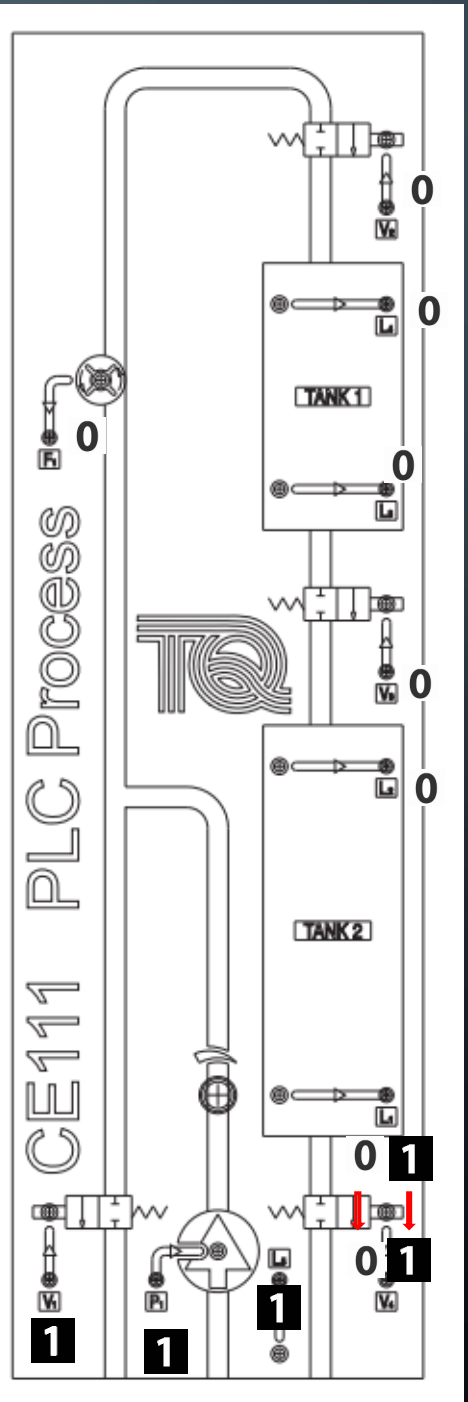




State B:
X006 was just turned ON and T1 started the count-down process

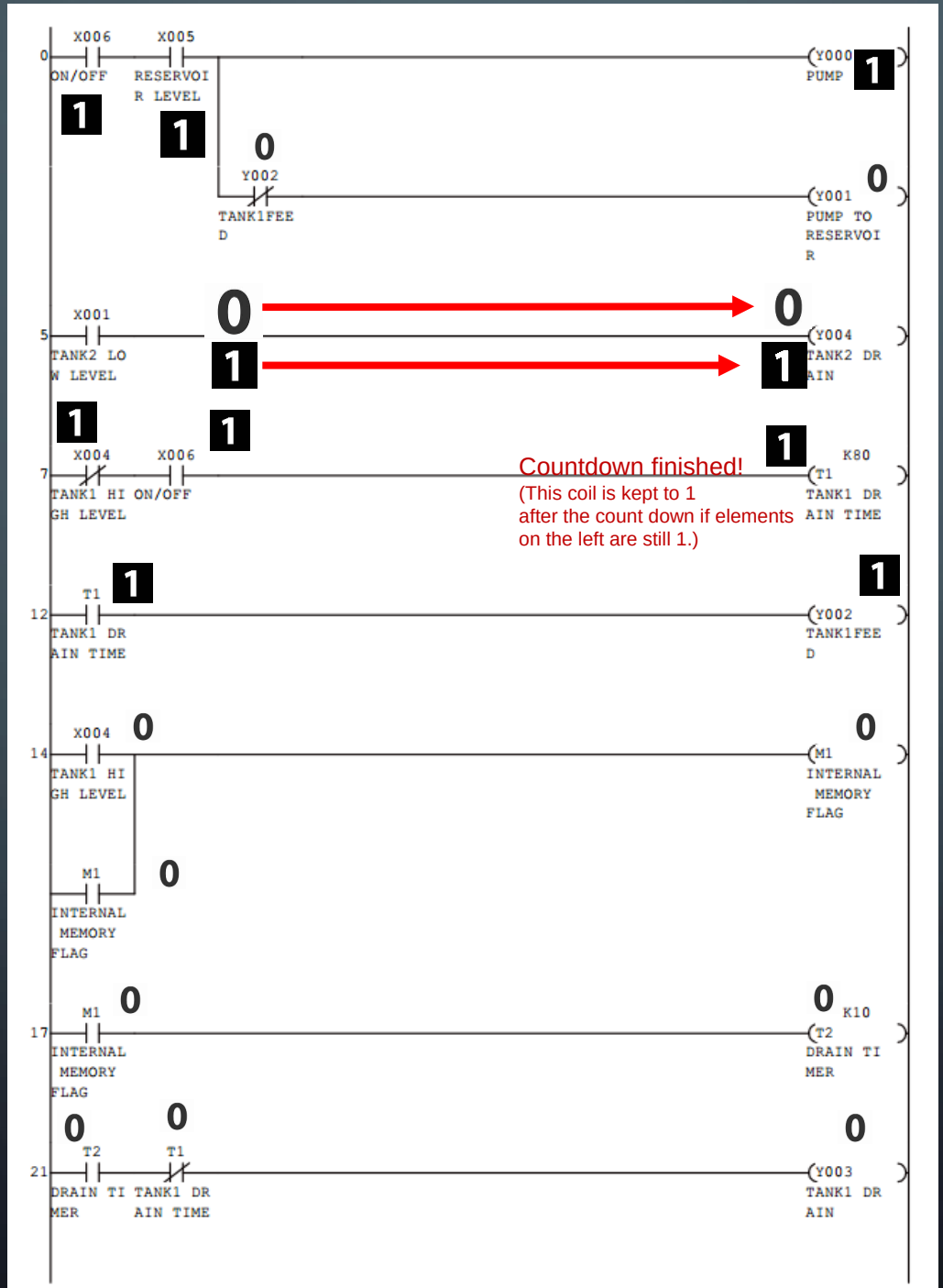
- Assume that there is no water in either tank.

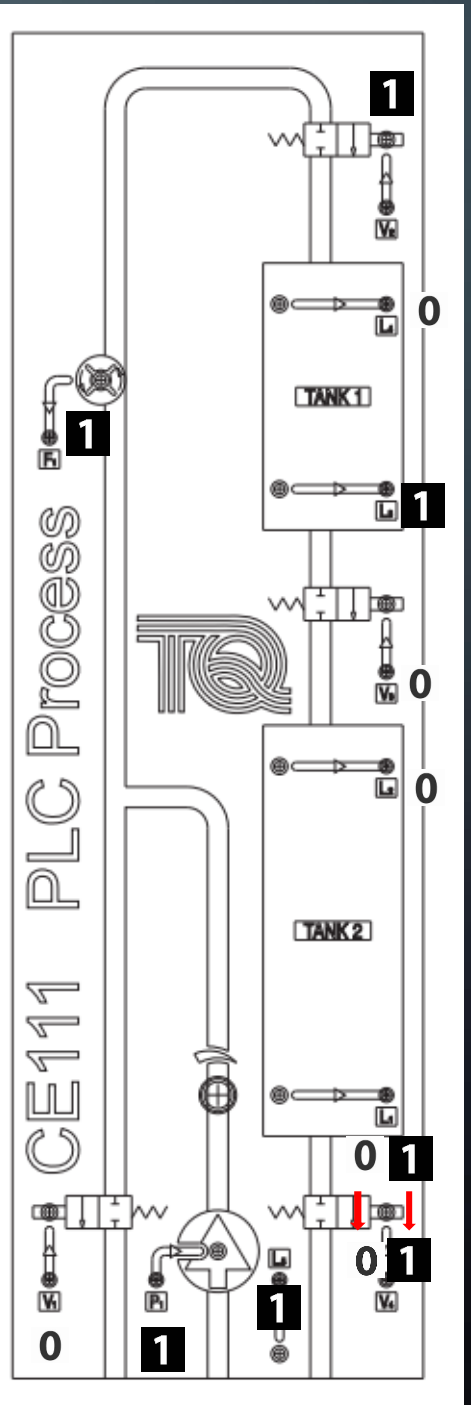




State C:
T1 just finished the count-down process and entered the feeding tank1 process

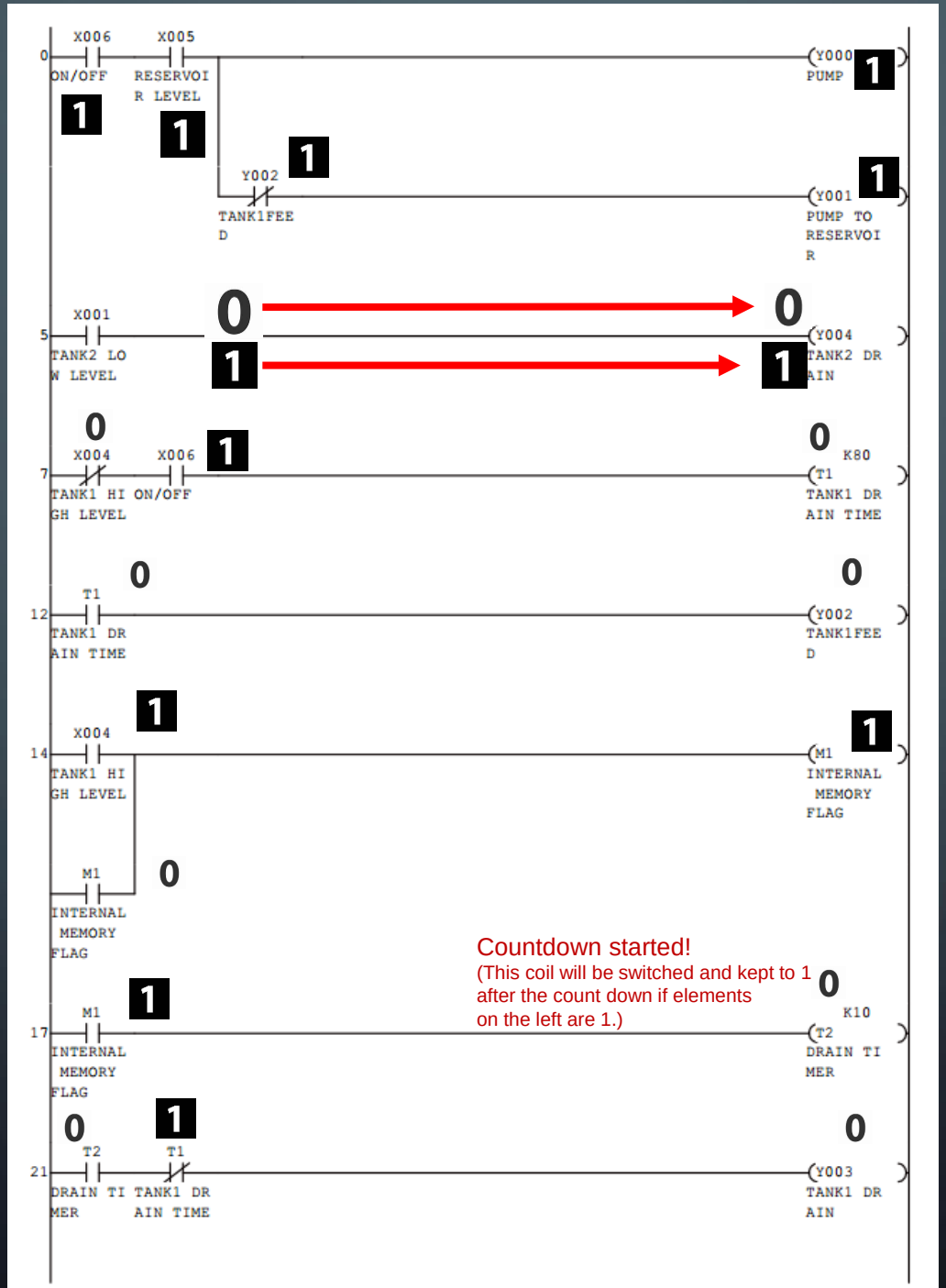
- Tank 1 is not full yet
- X006 keeps in ON mode

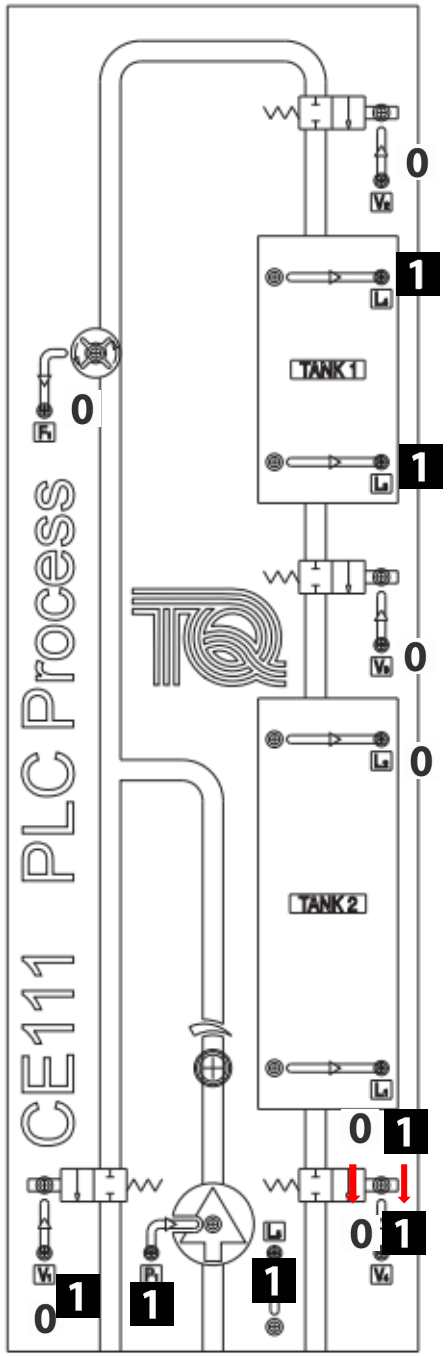




State D:
Water in Tank 1 had just reached the high level

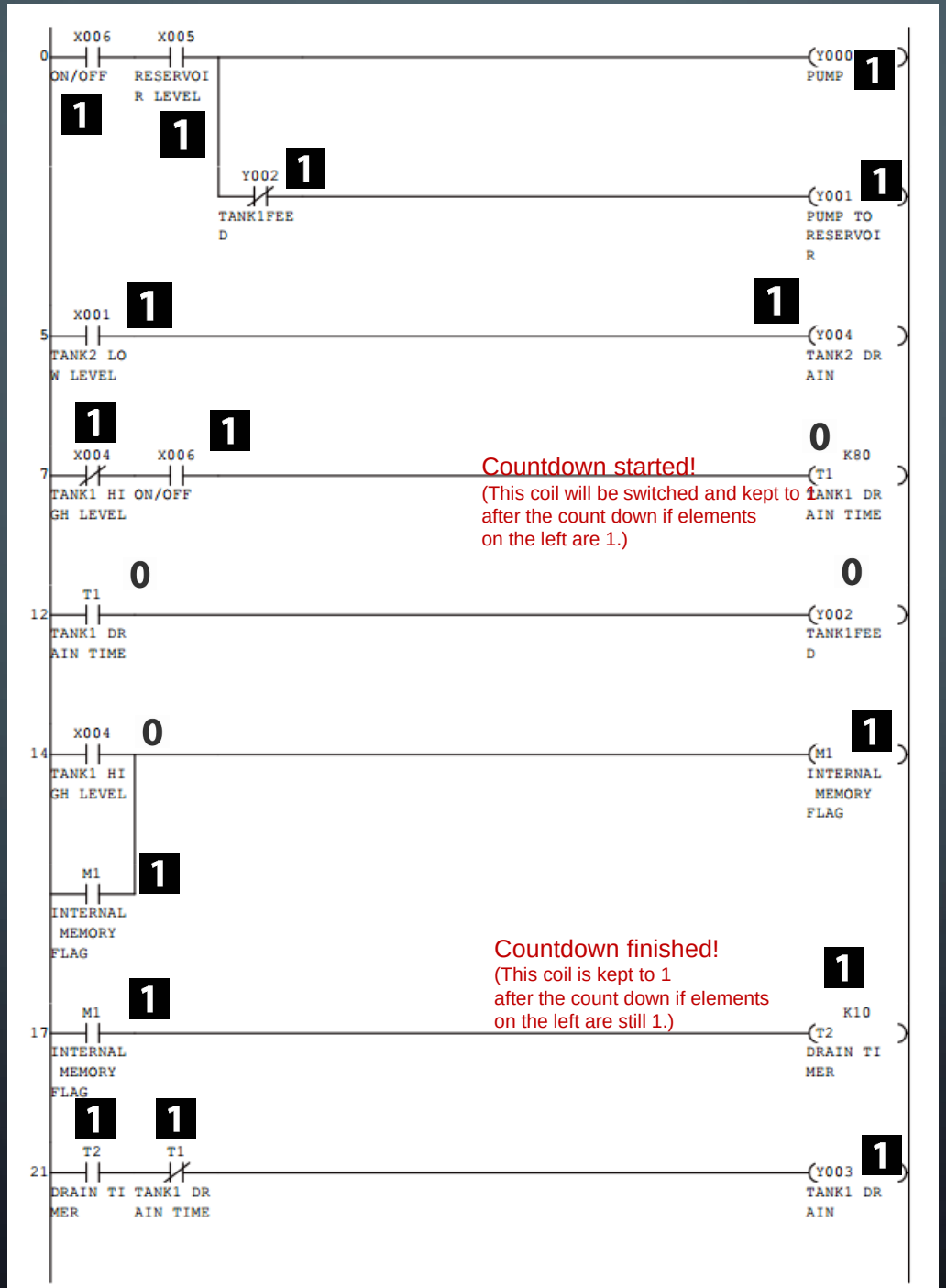
- The feed stopped immediately
- T2 enters count-down process

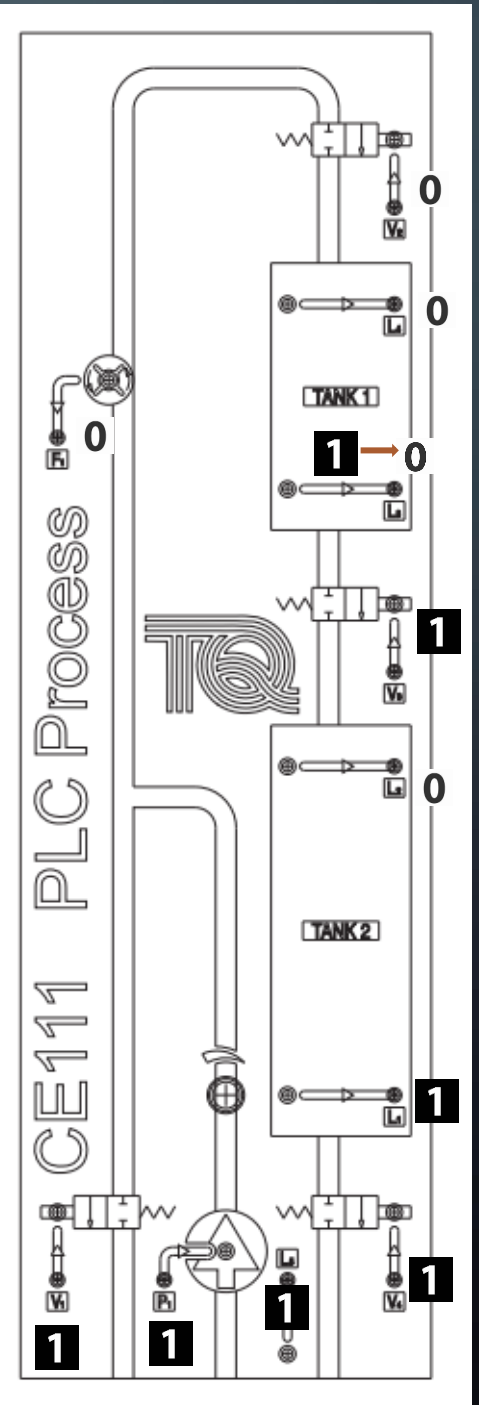




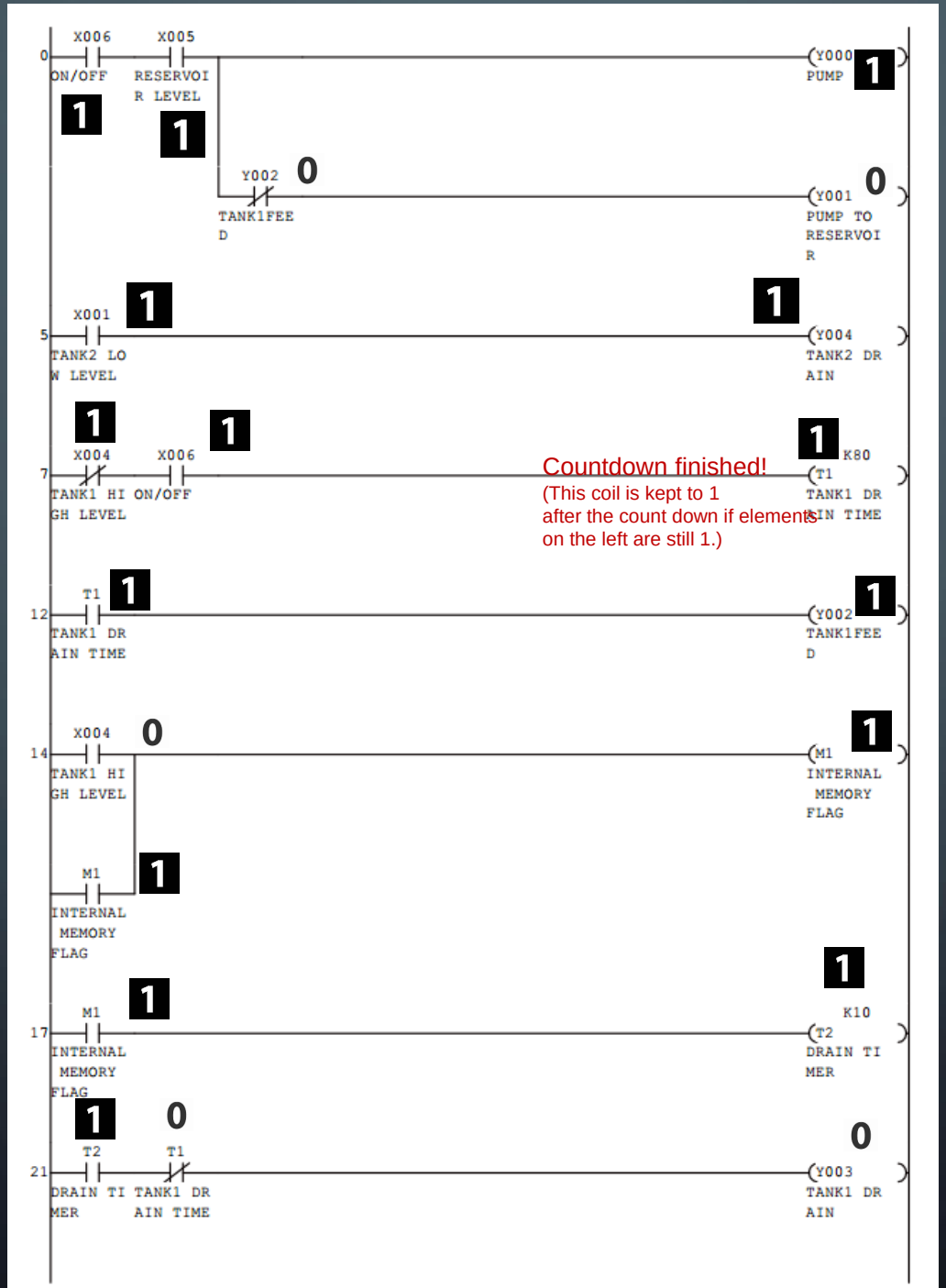
State E:
T2 finished count-down and Tank 1 started to drain

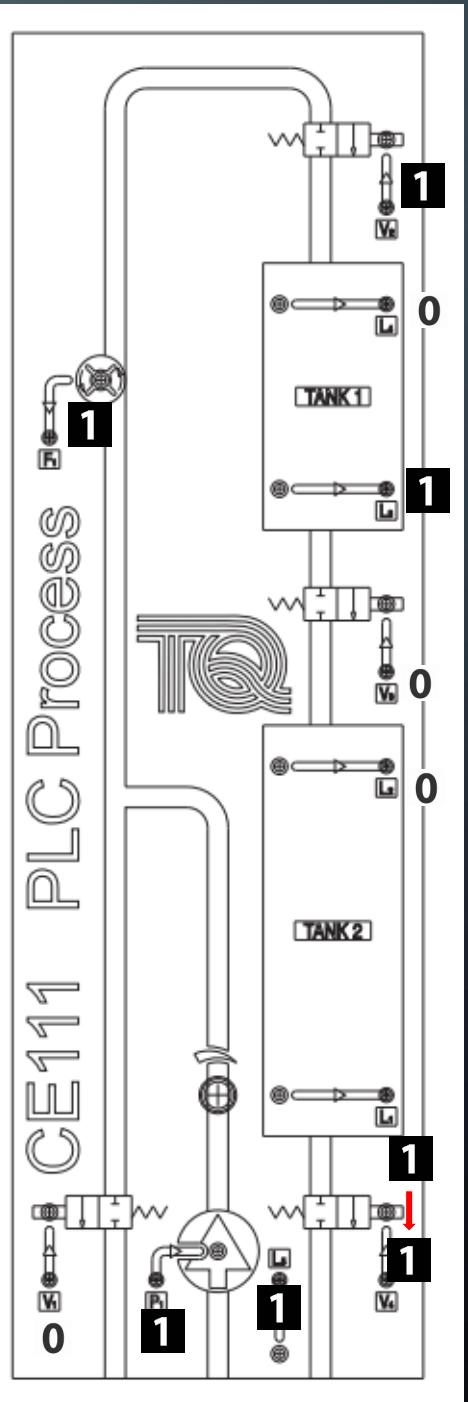
- Water in Tank 1 dropped lower than high level
- T1 entered count-down process





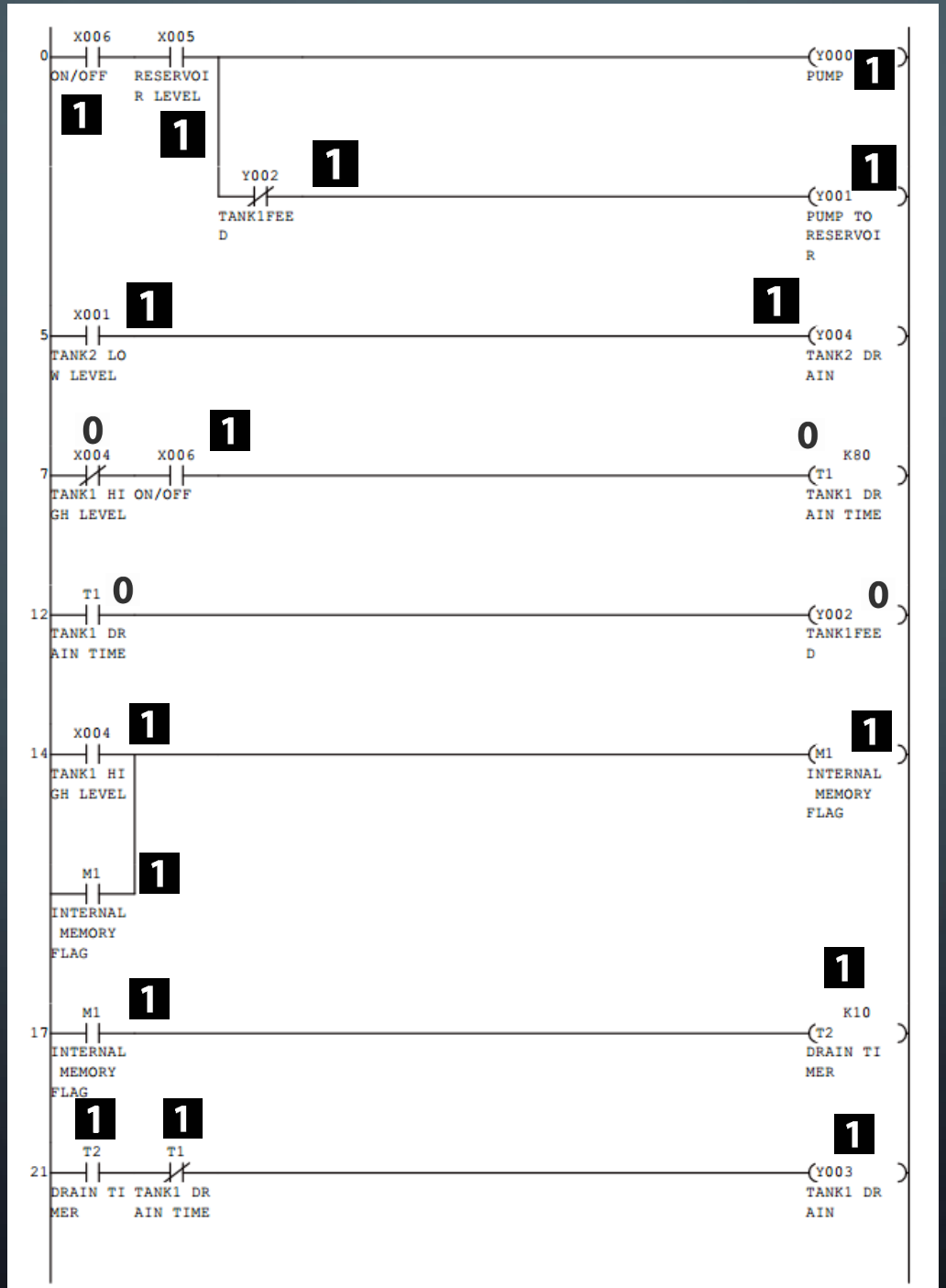
State F:
T1 finished count-down, drain stopped, and started to feed Tank 1

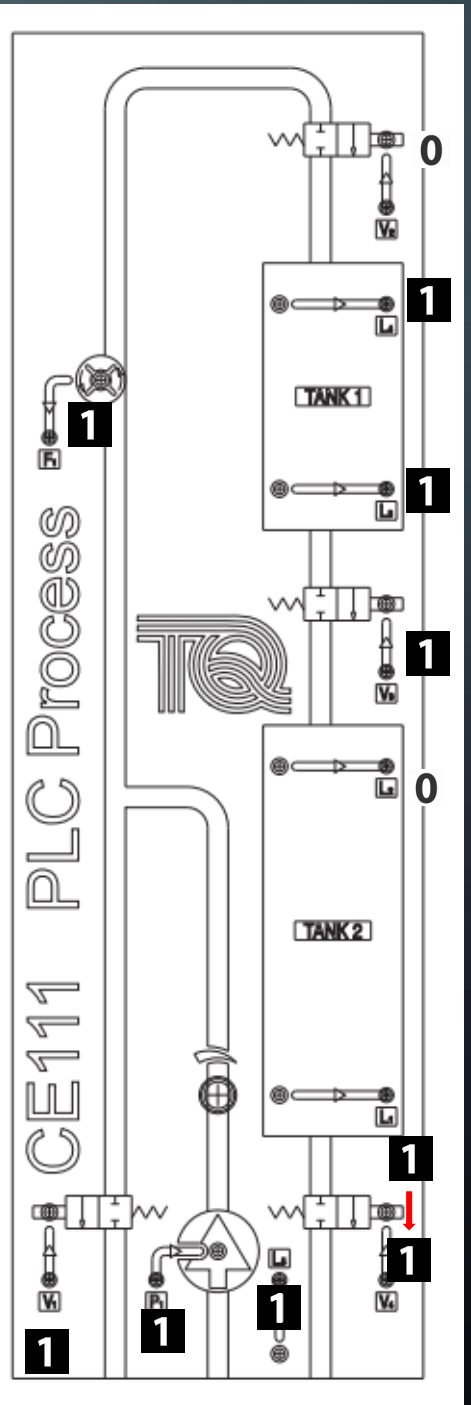




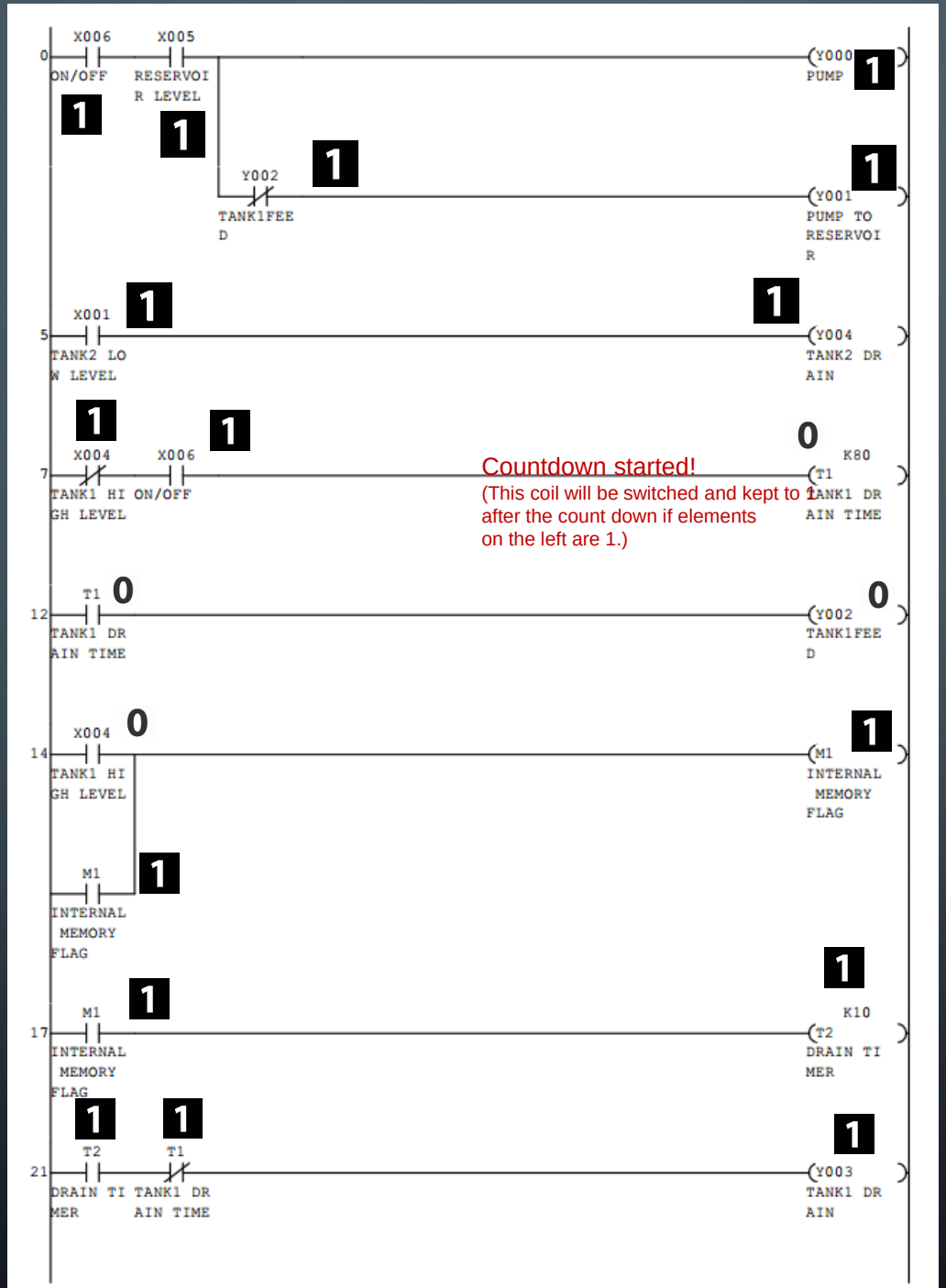
State G:
Water in Tank 1 had just reached the high level

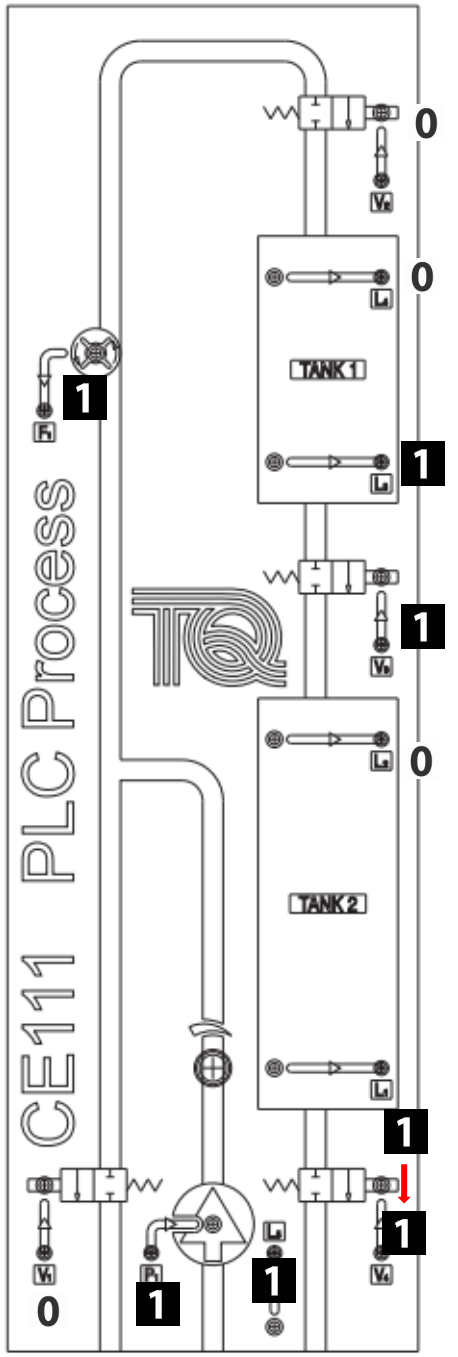
- The feed stopped immediately
- Timer1 reset to zero
- Tank1 Drain just started



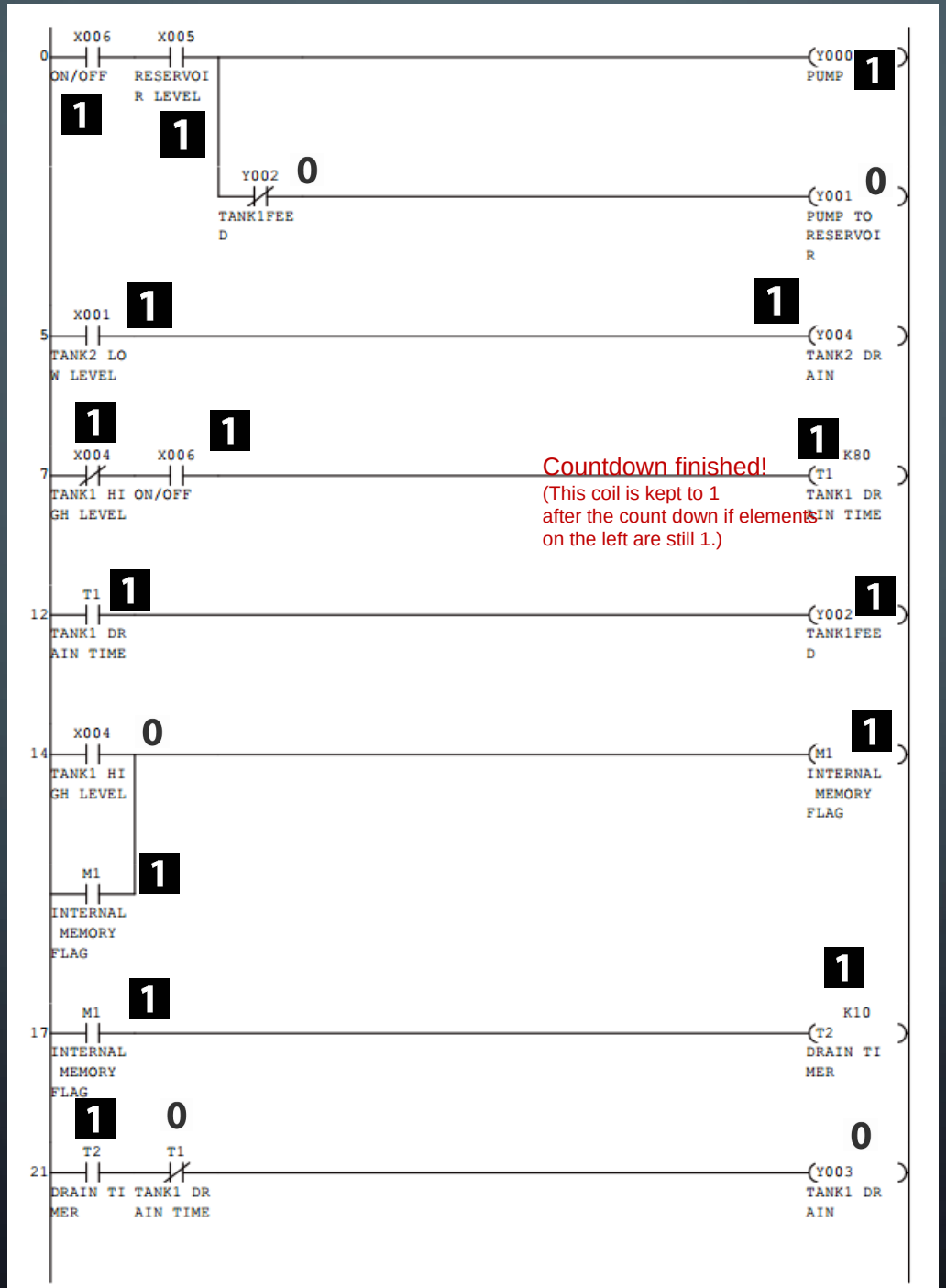


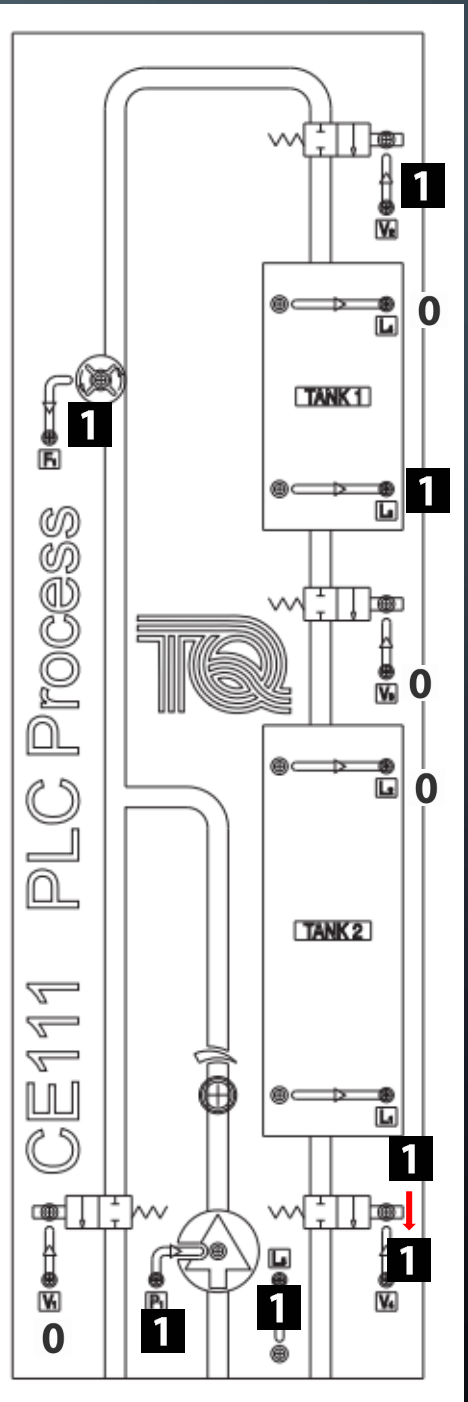
State H:
Water in Tank 1 was draining from high level





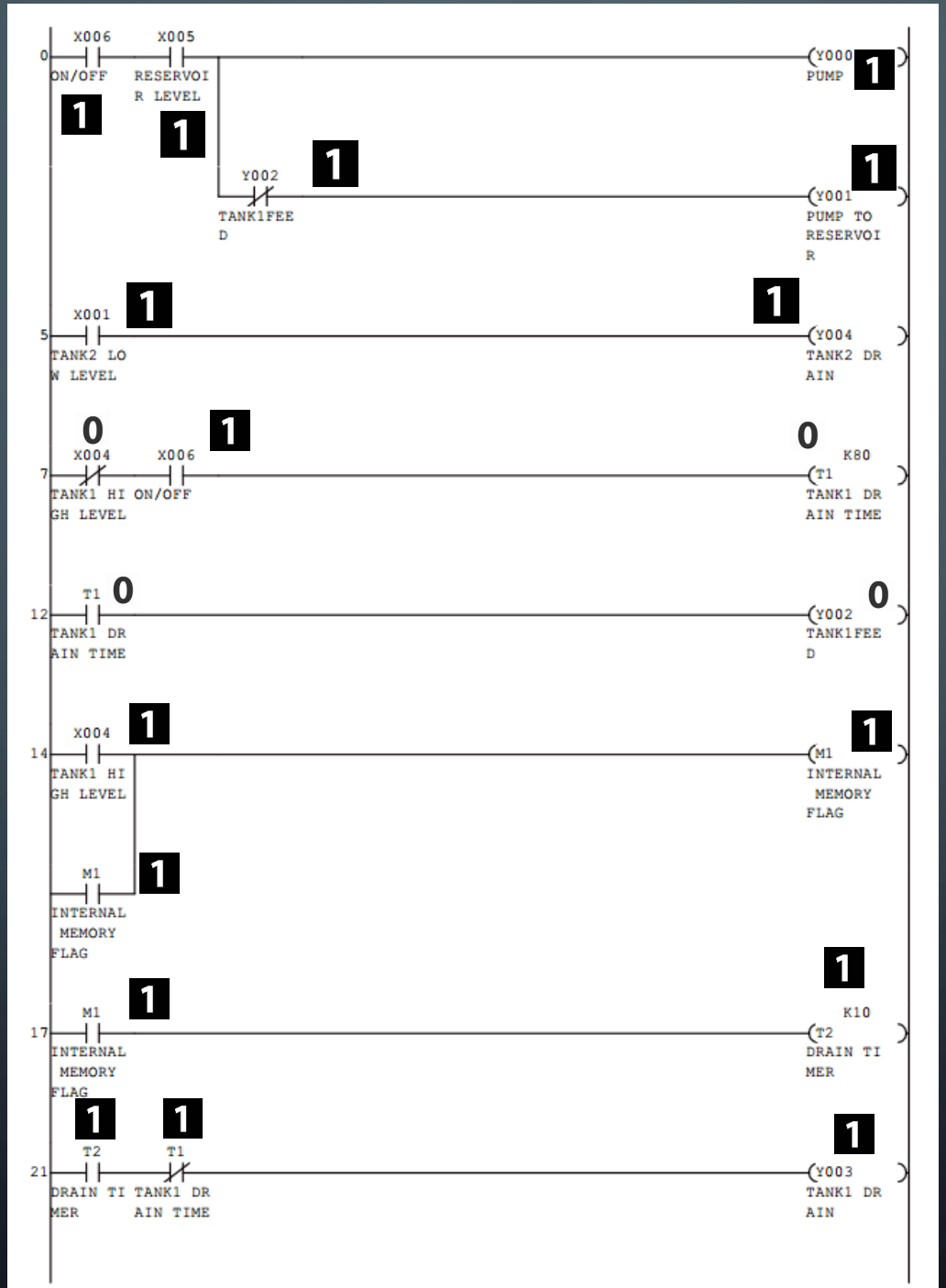
State F:
T1 finished count-down, drain stopped, and started to feed Tank 1

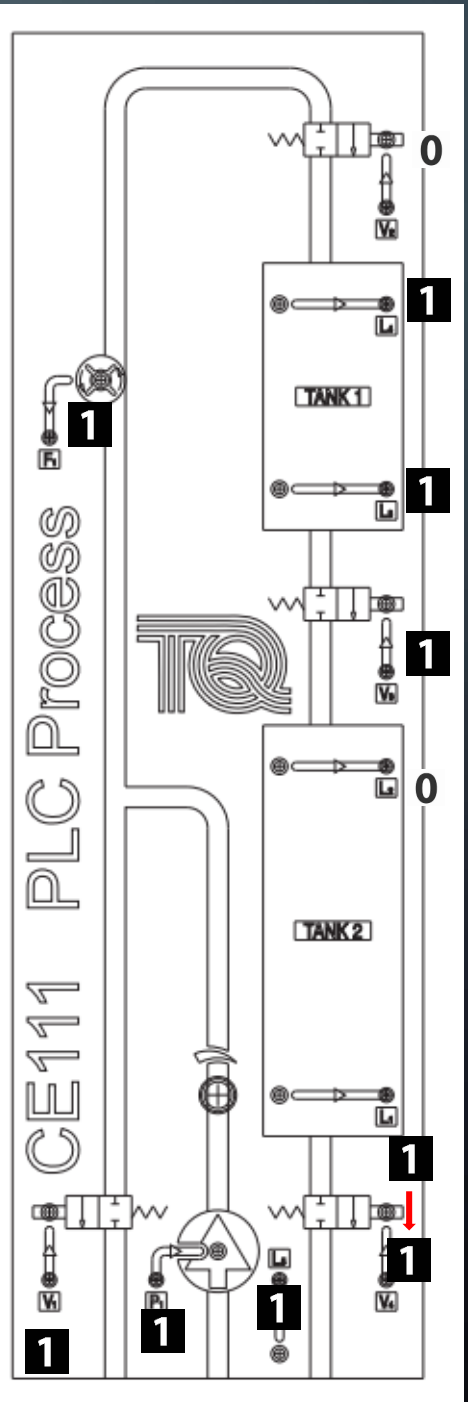




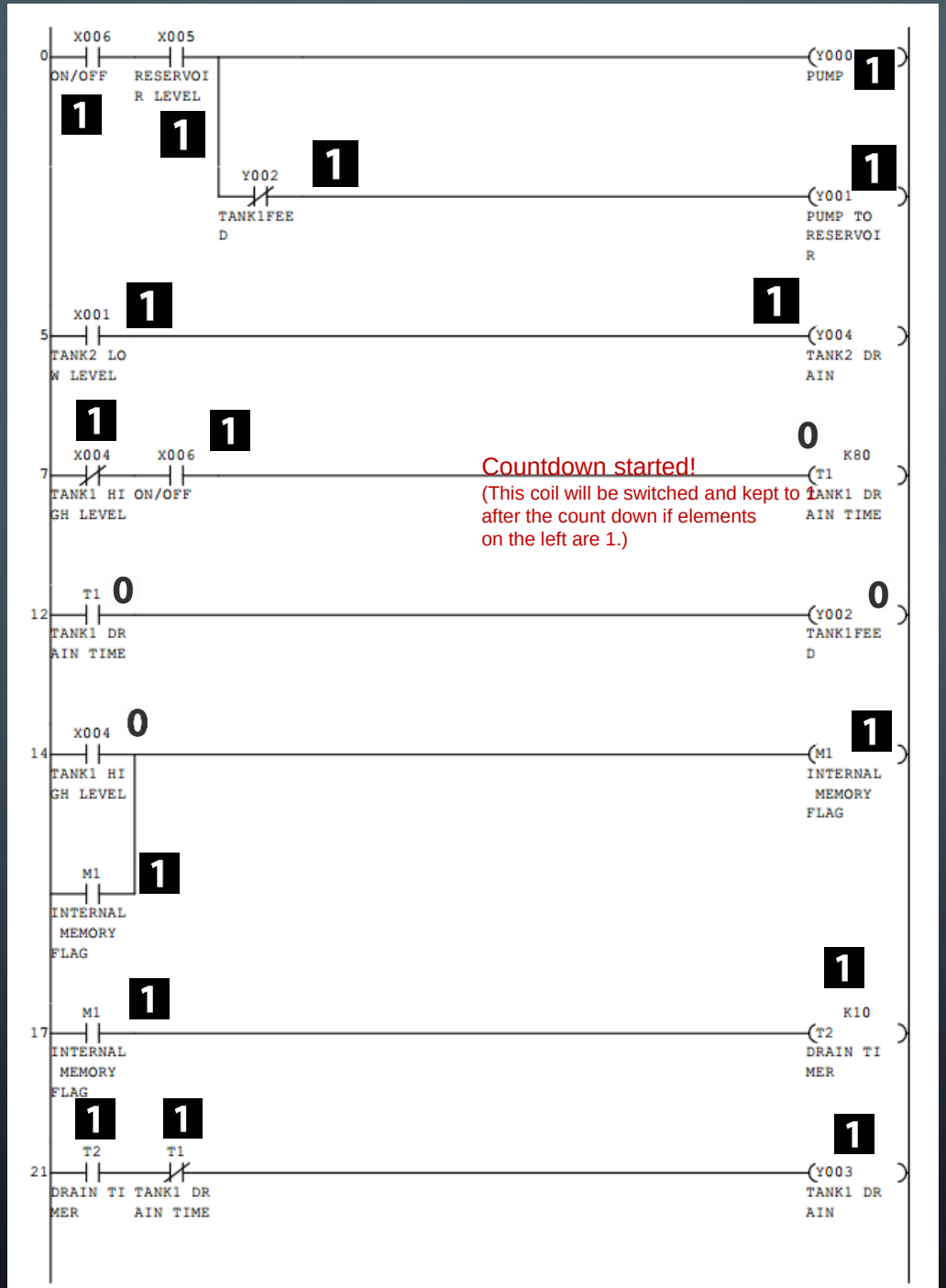
State G:
Water in Tank 1 had just reached the high level

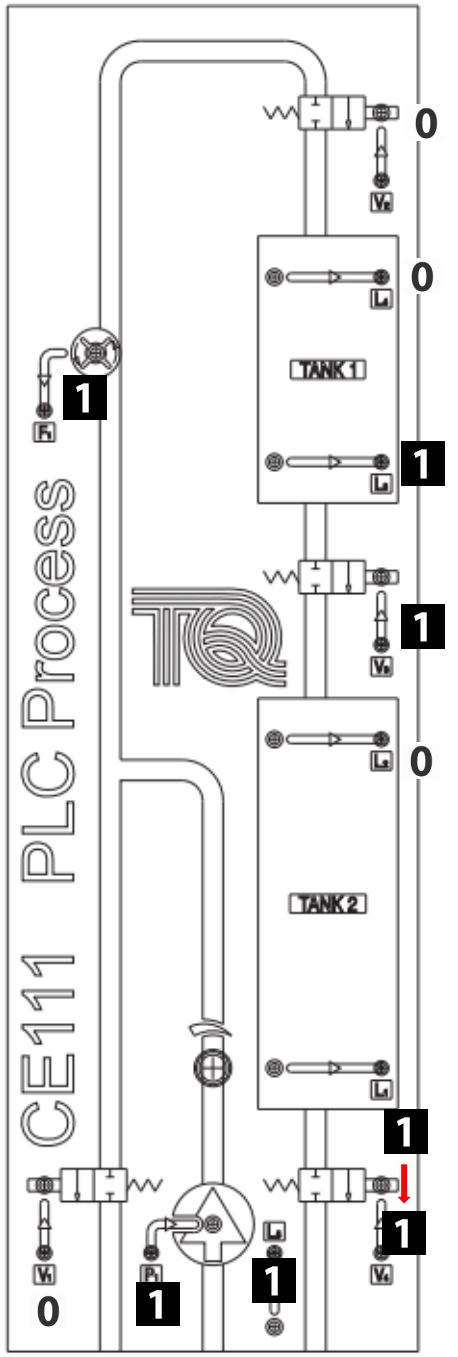
- The feed stopped immediately
- Timer1 reset to zero
- Tank1 Drain just started



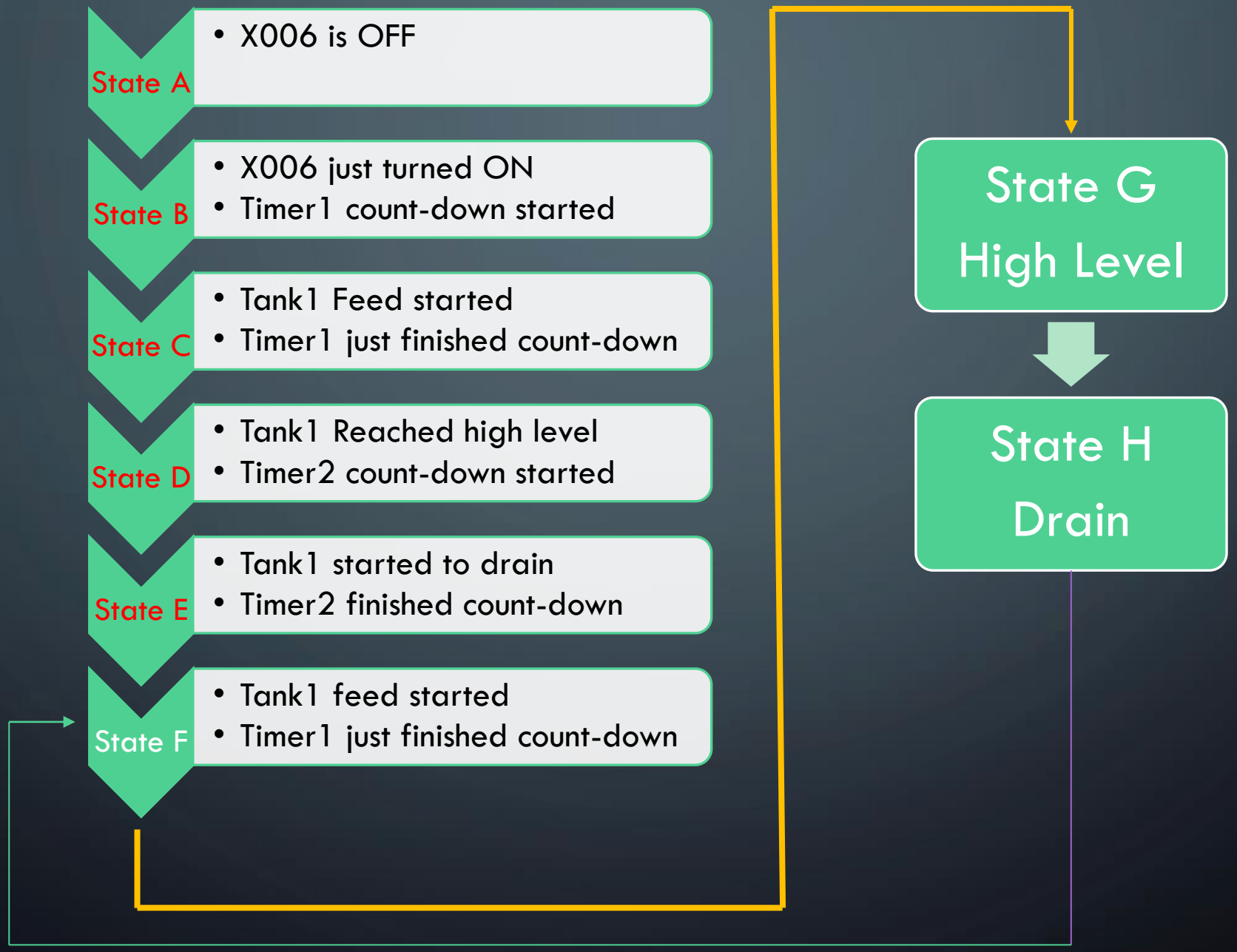


State H:
Water in Tank 1 was draining from high level





KEEP LOOPING IN STATES OF F G H


State A
• X006 is OFF
State B
• X006 just turned ON
• Timer1 count-down started
State C
• Tank1 Feed started
• Timer1 just finished count-down
State D
• Tank1 Reached high level
• Timer2 count-down started
State E
• Tank1 started to drain
• Timer2 finished count-down
State F
• Tank1 feed started
• Timer1 just finished count-down
State G
High Level
State H
Drain