改善
KAIZEN
TEAM

Automation Slides With Notes

(2nd Semester 2023/2024)
Notes are written by Nada Ababneh

Automation

Chapter Nine:

Discrete Control Using Programmable Logic Controllers

Dr. Eng. Baha’eddin Alhaj Hasan
Department of Industrial Engineering

Chapter 9: Discrete Control

Sections:

1. Discrete Process Control
2. Ladder Logic Diagrams
3. Programmable Logic Controllers

→ PLC → (for discrete controllers)

2 States → 0 → ex: Low/off/.....
2 States → 1 → ex: High/on/.....

* غالباً :-

0 → zero volt
1 → five volt

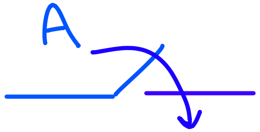
Sec 9.1: Discrete Process Control

- Continuous Control: deals with controlling continuous variables or parameters in the system.

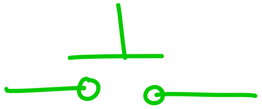
- Discrete Control: Control systems that operate on parameters and variables that change at discrete moments in time or at discrete events,
  - usually binary (0 or 1, off or on, open or closed, etc.)
  - Called also: switching systems.

Sensors and Actuators Used in Discrete Process Control

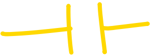
Sensors	Interpretation	Actuators	Interpretation
Limit switch	Contact/no contact	Motor ex: stepper motor	On/off
Photo-detector	On/off	Valve	Open/closed
Timer	On/off	Clutch	Engaged/not engaged
Push-button switch	On/off	Control relay	Contact/no contact
Control relay	Contact/no contact	Light	On/off
Circuit breaker	Contact/no contact	Solenoid	Energized/not energized



Normally Open switch (NO)



No pushbutton

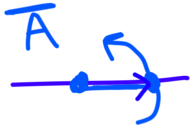


No switch

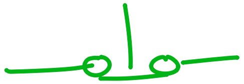
NO

normally open

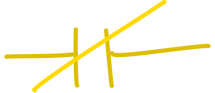
Push button



Normally Closed switch (NC)



NC push buttom switch



NC switch

N C

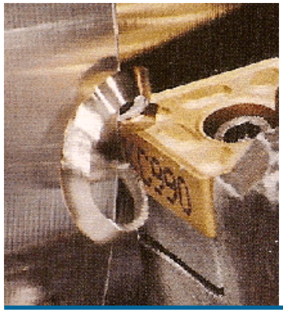
normally closed

Push button

Categories of Discrete Control

- Logic control: event-driven changes
- Sequencing: time-driven changes

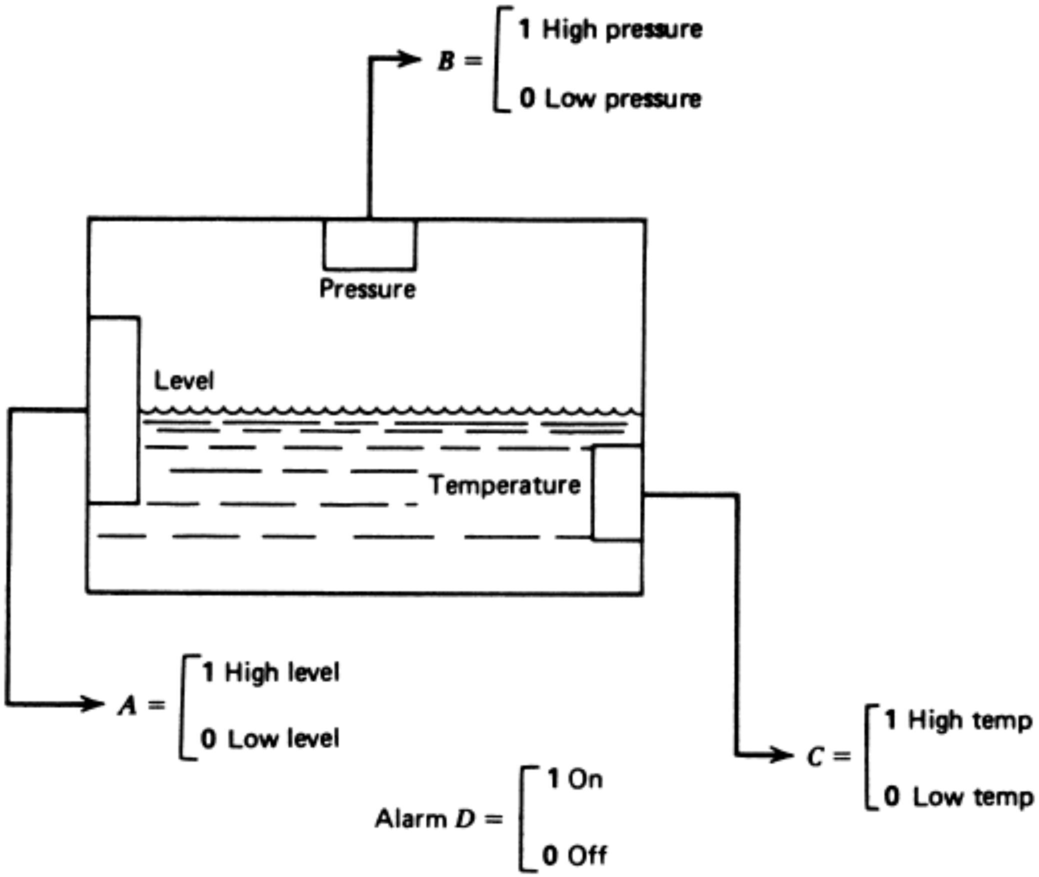
- **Logic Control:** a switching system whose output at any moment is determined exclusively by the values of inputs.
  - No memory
  - No operating characteristics that depend on time
  - Also called *combinational logic control*



Digital Variables

2 States → 0 → ex: Low/off/.....
→ 1 → ex: High/on/.....

* غالباً :-
0 → zero volt
1 → five volt

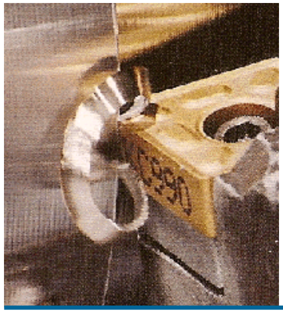


$B =$ 1 High pressure / 0 Low pressure

$A =$ 1 High level / 0 Low level

$C =$ 1 High temp / 0 Low temp

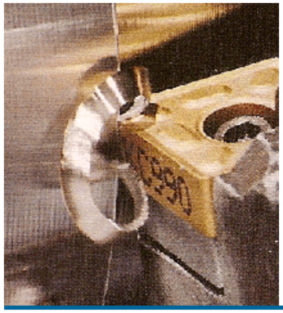
Alarm $D =$ 1 On / 0 Off



Boolean Algebra

The alarm will be triggered when the Boolean variable *D* goes to the logic true state. The alarm conditions are:

1. Low level with high pressure
2. High level with high temperature
3. High level with low temperature and high pressure



Boolean Algebra

A (Level)

B (Pressure)

C (Temperature)

D (Alarm)

We now define a Boolean expression with AND operations that will give a $D = 1$ for each condition:

1. $D = \overline{A} \cdot B$ will give $D = 1$ for condition 1.
2. $D = A \cdot C$ will give $D = 1$ for condition 2.
3. $D = A \cdot \overline{C} \cdot B$ will give $D = 1$ for condition 3.

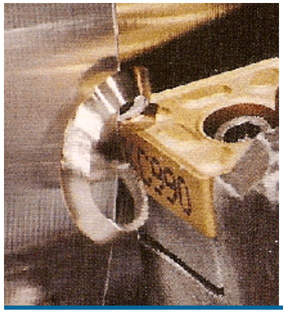
The alarm conditions are:

1. Low level with high pressure
2. High level with high temperature
3. High level with low temperature and high pressure

The final logic equation results from combining all three conditions so that if any is true, the alarm will sound ($D = 1$). This is accomplished with the OR operation

$$D = \overline{A} \cdot B + A \cdot C + A \cdot \overline{C} \cdot B \tag{2}$$

This equation would now form the starting point for a design of electronic digital circuitry that would perform the indicated operations.



Boolean Algebra

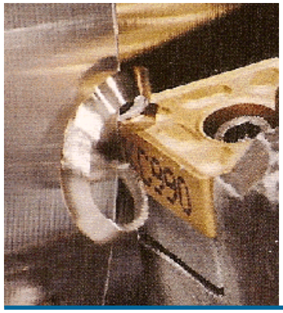
Develop a digital circuit using AND/OR gates that implements Equation (2).

Solution

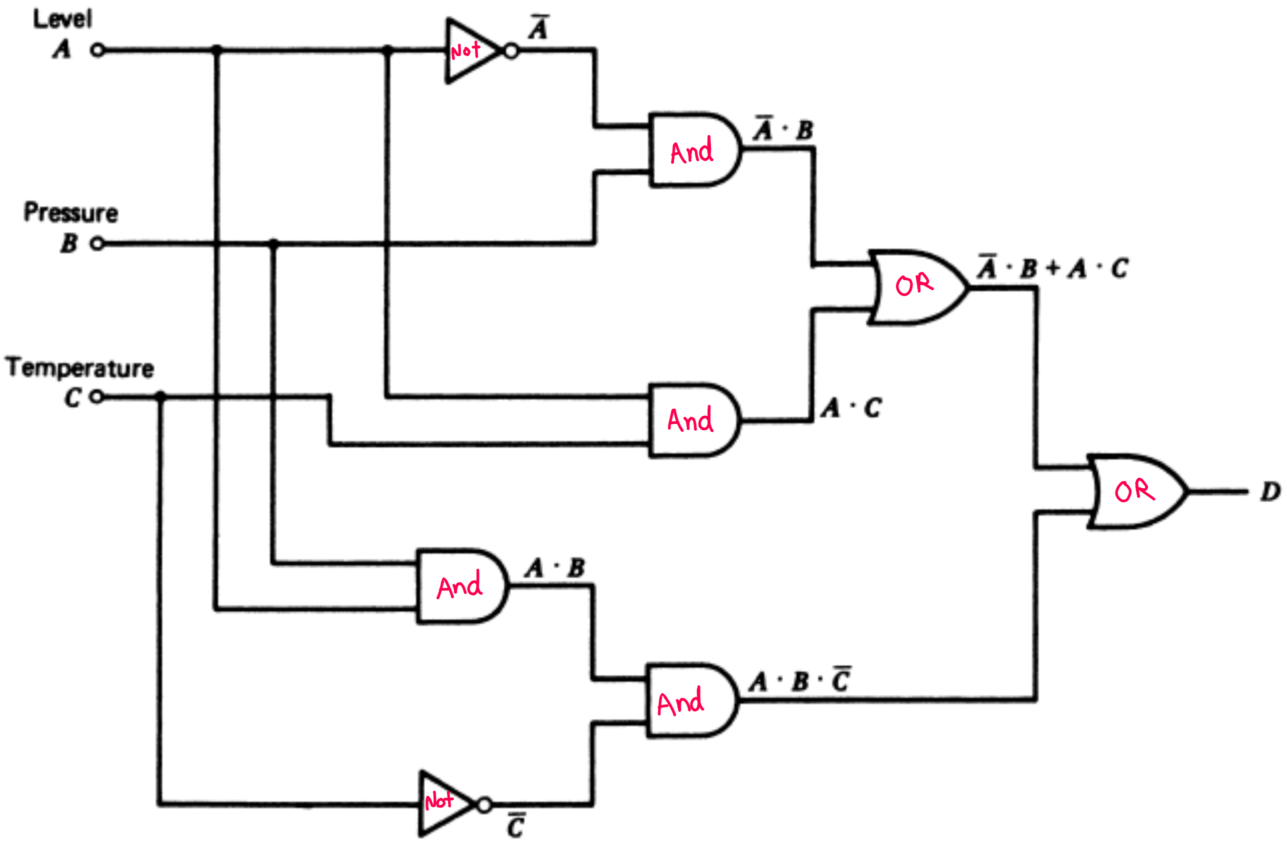
The problem posed in Section 2.3 (with Figure 1) has a Boolean equation solution of

$$D = \overline{A} \cdot B + A \cdot C + A \cdot \overline{C} \cdot B \qquad \textbf{(2)}$$

$$\boxed{D = \overline{A} \cdot B + A \cdot C + A \cdot \overline{C} \cdot B}$$



Boolean Algebra



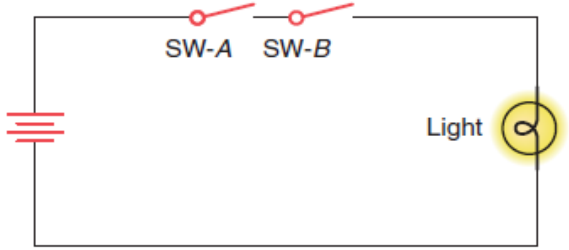
©201

Fundamentals of Logic

-Single output < 0 / 1

-multiple input
(not ما عدا ال)

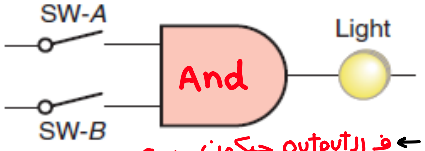
Hardwired circuit



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)

Logic representation

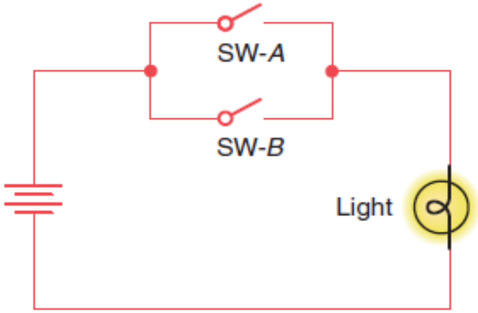


(And)

لو أي input ← 0 ← ف الoutput حيكون ← 0

لو كلهم شغالين يعني جميع الinput ← 1 ← حيكون الoutput ← 1

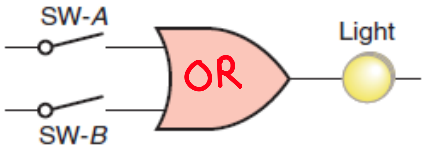
Hardwired circuit



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

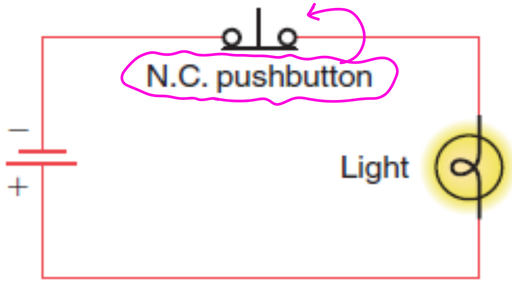


(OR)

اي input ← 1

راح يعطيني output ← 1

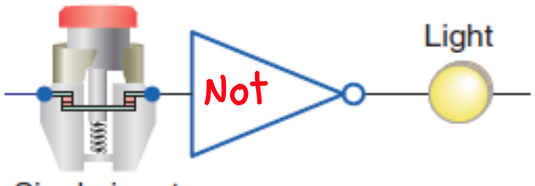
Hardwired circuit



Truth table

Pushbutton		Light	
Not pressed	(0)	On	(1)
Pressed	(1)	Off	(0)

Logic representation



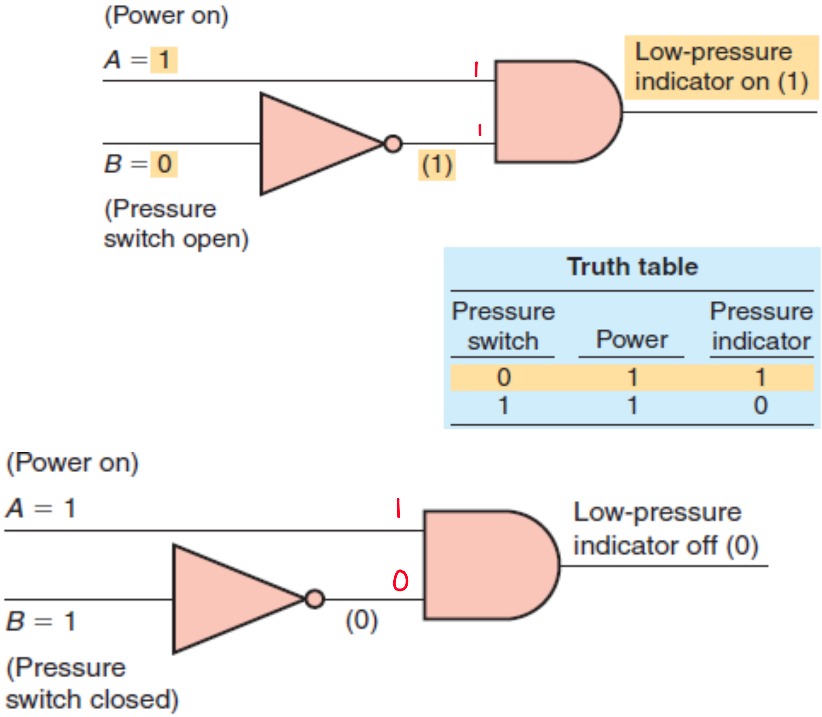
Single input

(Not)

عكس ال input

N.C A normally closed pushbutton.

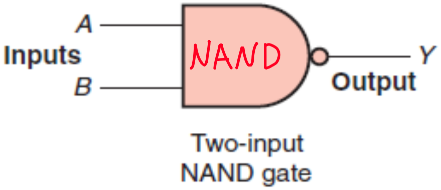
Fundamentals of Logic



Truth table

Pressure switch	Power	Pressure indicator
0	1	1
1	1	0

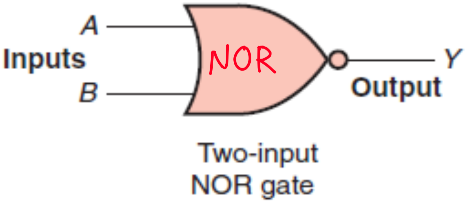
Figure 4-11 NOT function is most often used in conjunction with an AND gate.



NAND truth table

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

Figure 4-12 NAND gate symbol and truth table.



NOR truth table

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

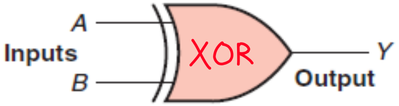
Figure 4-13 NOR gate symbol and truth table.

(XOR)

(XNOR) هي عكسها

input متشابه → output zero

input مختلف → output one

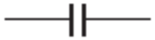
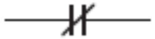
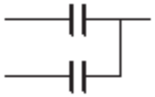
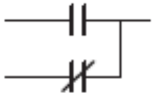
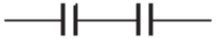
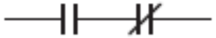
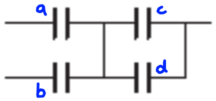
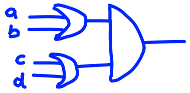
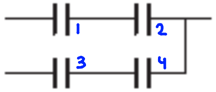
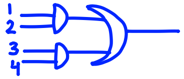
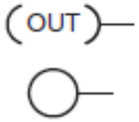
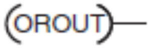
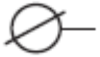


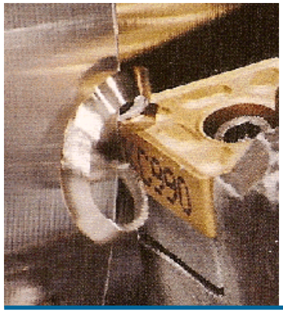
Truth table

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

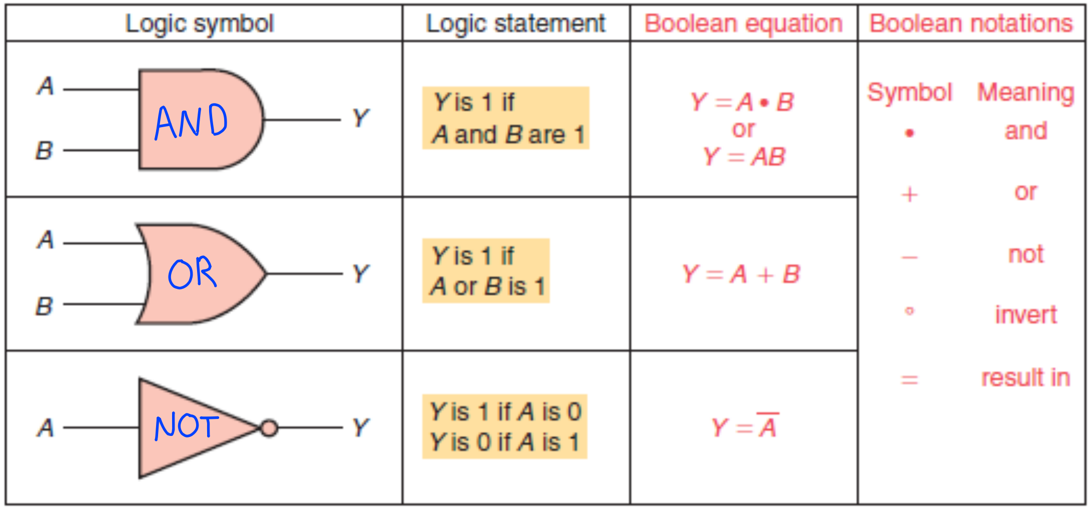
Figure 4.14 The XOR gate symbol and truth table.

Table 4-1 Typical Boolean Instruction or Statement List

Boolean Instruction and Function	Graphic Symbol
Store (STR)–Load (LD) → output Begins a new rung or an additional branch in a rung with a normally open contact.	
Store Not (STR NOT)–Load Not (LD NOT) Begins a new rung or an additional branch in a rung with a normally closed contact.	
Or (OR) Logically ORs a normally open contact in parallel with another contact in a rung.	
Or Not (OR NOT) Logically ORs a normally closed contact in parallel with another contact in a rung.	
And (AND) Logically ANDs a normally open contact in series with another contact in a rung.	
And Not (AND NOT) Logically ANDs a normally closed contact in series with another contact in a rung.	
And Store (AND STR)–And Load (AND LD) Logically ANDs two branches of a rung in series.	 
Or Store (OR STR)–Or Load (OR LOAD) Logically ORs two branches of a rung in parallel.	 
Out (OUT) Reflects the status of the rung (on/off) and outputs the discrete (ON/OFF) state to the specified image register point or memory location.	
Or Out (OR OUT) Reflects the status of the rung and outputs the discrete (ON/OFF) state to the image register. Multiple OR OUT instructions referencing the same discrete point can be used in the program.	
Output Not (OUT NOT) Reflects the status of the rung and turns the output OFF for an ON execution condition; turns the output ON for an OFF execution condition.	

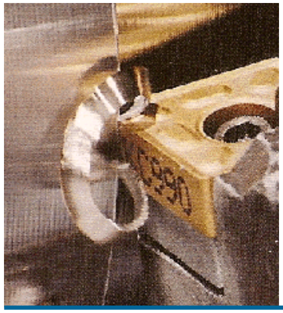


Fundamentals of Logic



Logic symbol	Logic statement	Boolean equation	Boolean notations
A B AND Y	*Y* is 1 if *A* and *B* are 1	$Y = A \bullet B$ or $Y = AB$	Symbol Meaning • and + or ‾ not ∘ invert = result in
A B OR Y	*Y* is 1 if *A* or *B* is 1	$Y = A + B$	
A NOT Y	*Y* is 1 if *A* is 0 *Y* is 0 if *A* is 1	$Y = \overline{A}$	

Figure 4-15 Boolean algebra as related to AND, OR, and NOT functions.



Fundamentals of Logic

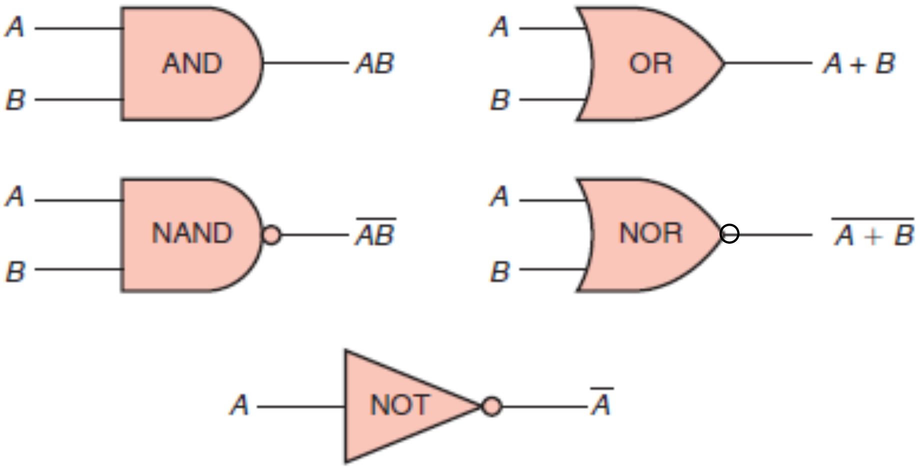
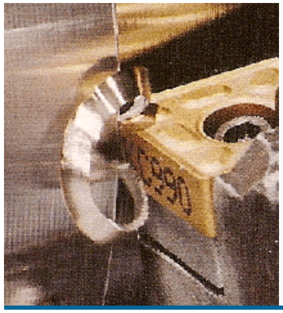
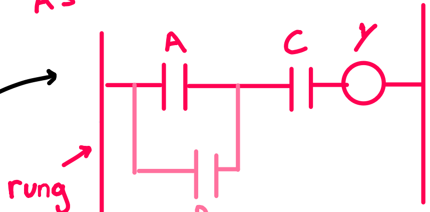


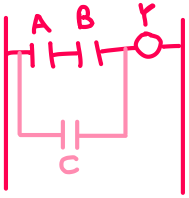
Figure 4-16 Logic operators used singly to form logical statements.

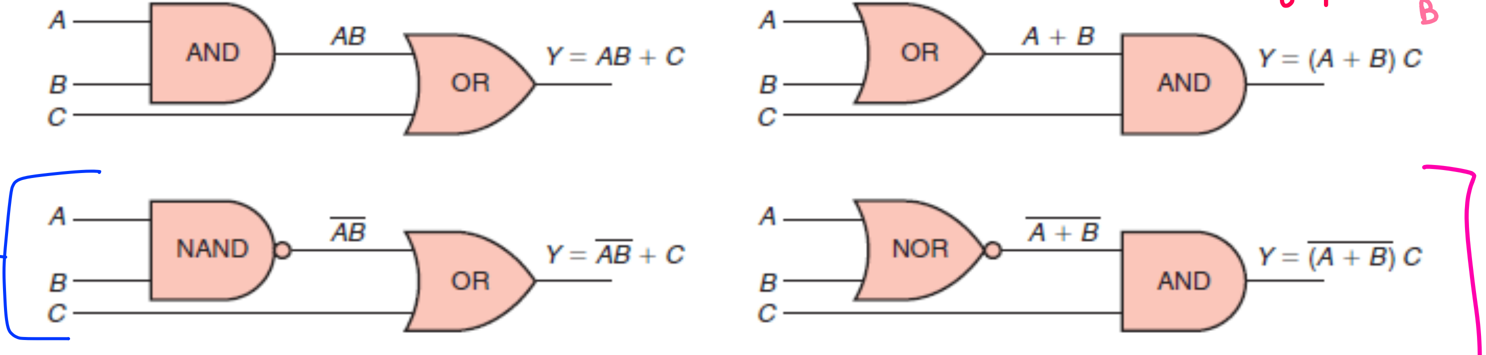


Fundamentals of Logic

As a Lader diagram

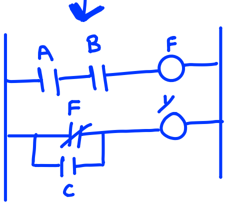


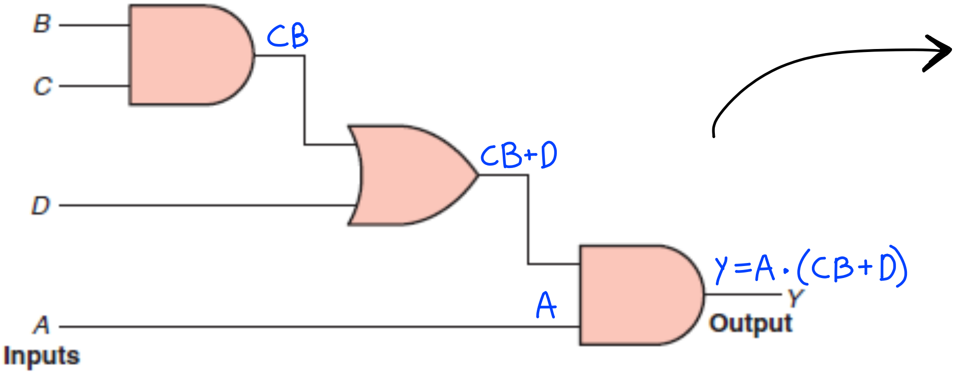


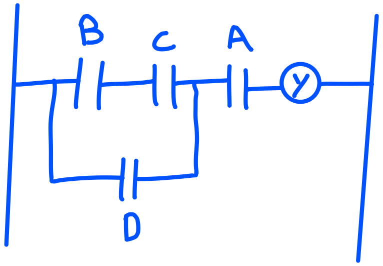


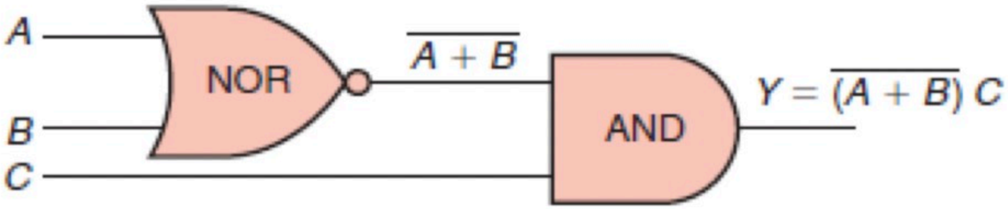
* in the next slide !!

Figure 4-17 Logic operators used in combination to form Boolean equations.

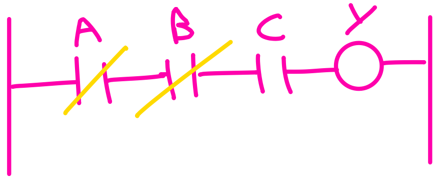








* first method



Demorgons Theory

$Y = \overline{(A+B)} . C$

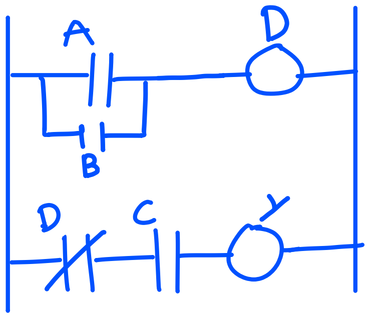
$= \bar{A} . \bar{B} . C$

* Second Method

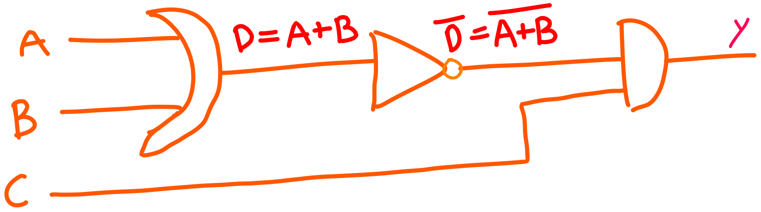
$Y = \overline{(A+B)}\, C$

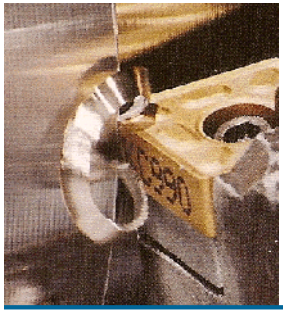
$A + B = D$

$Y = \bar{D} C$

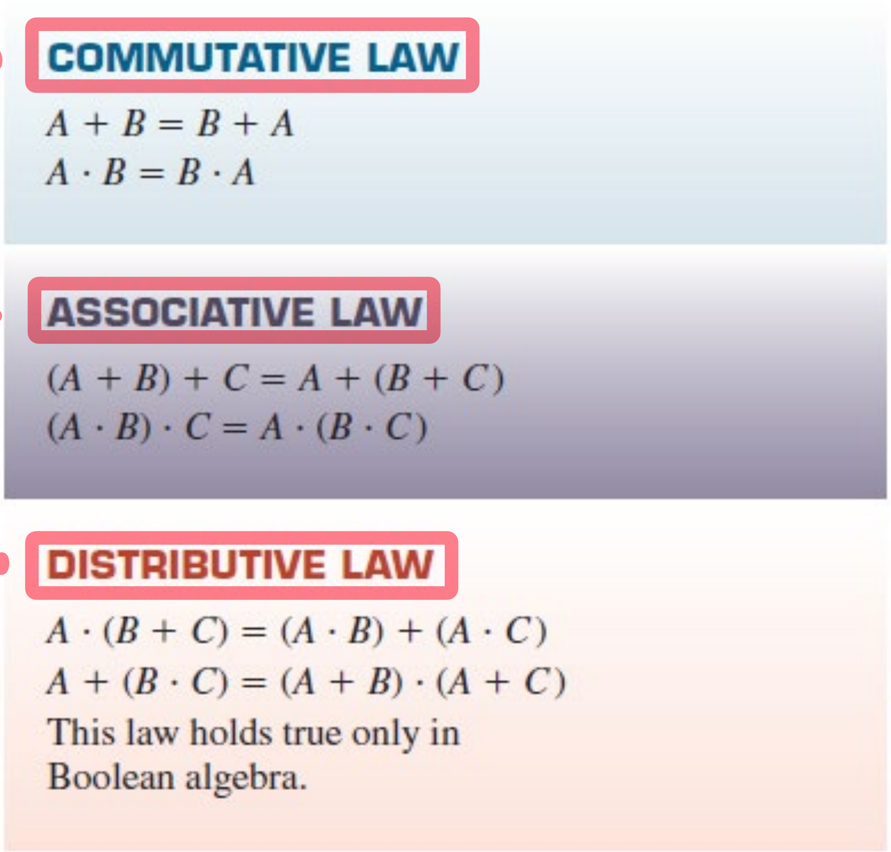


$Y = \overline{(A+B)}\, C$





Fundamentals of Logic



COMMUTATIVE LAW

$A + B = B + A$

$A \cdot B = B \cdot A$

ASSOCIATIVE LAW

$(A + B) + C = A + (B + C)$

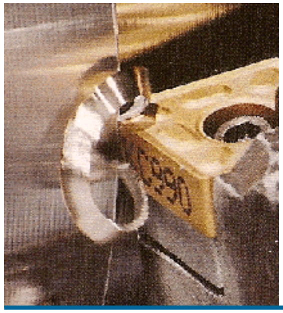
$(A \cdot B) \cdot C = A \cdot (B \cdot C)$

DISTRIBUTIVE LAW

$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

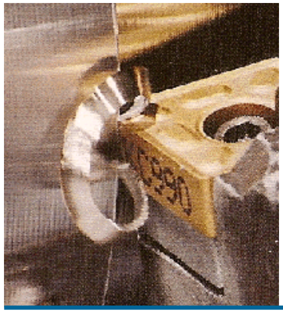
$A + (B \cdot C) = (A + B) \cdot (A + C)$

This law holds true only in Boolean algebra.

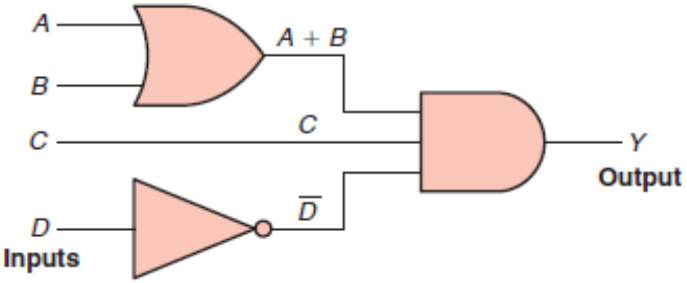


Fundamentals of Logic

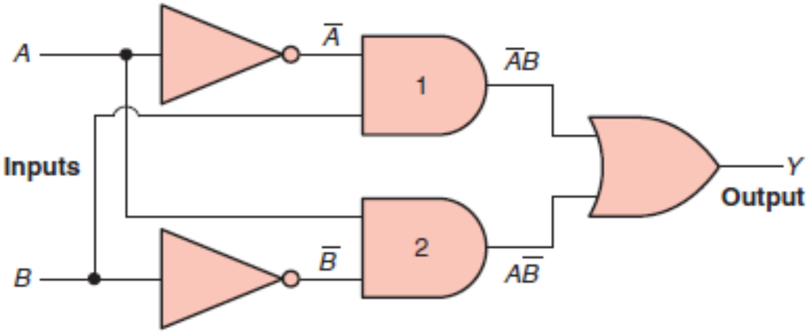
Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + (BC) = (A + B)(A + C)$	$A(B + C) = AB + AC$
* Absorption law	$A(A + B) = A$ *	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$



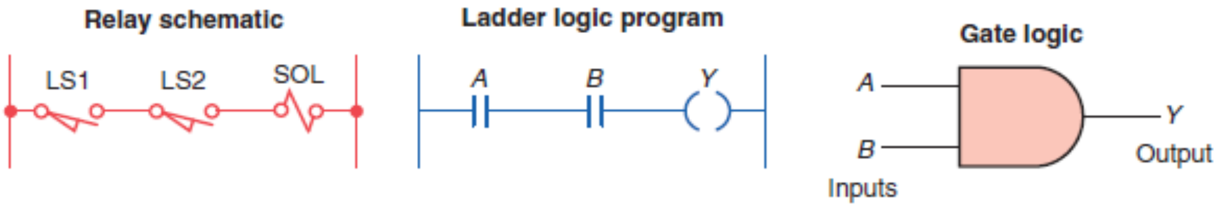
Fundamentals of Logic



$Y = (A+B).C.\overline{D}$



$Y = \overline{A}B + A\overline{B}$

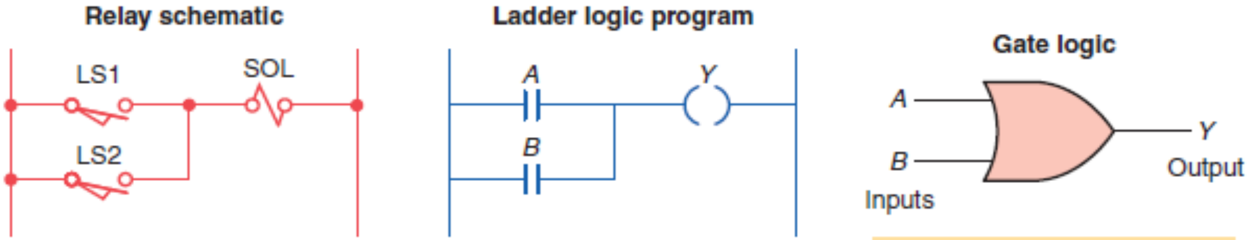


Relay schematic

Ladder logic program

Gate logic

Boolean equation: $AB = Y$

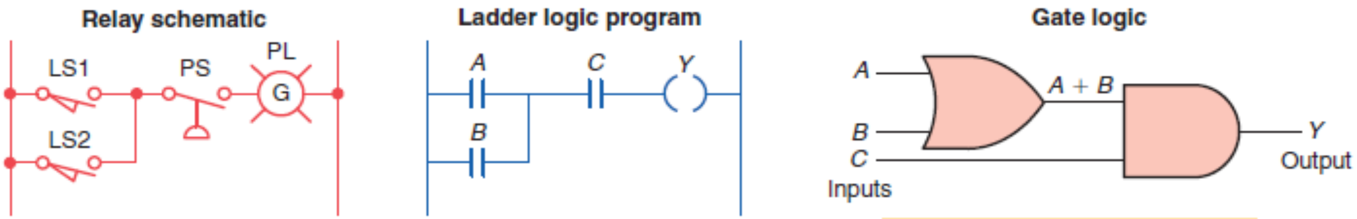


Relay schematic

Ladder logic program

Gate logic

Boolean equation: $A + B = Y$

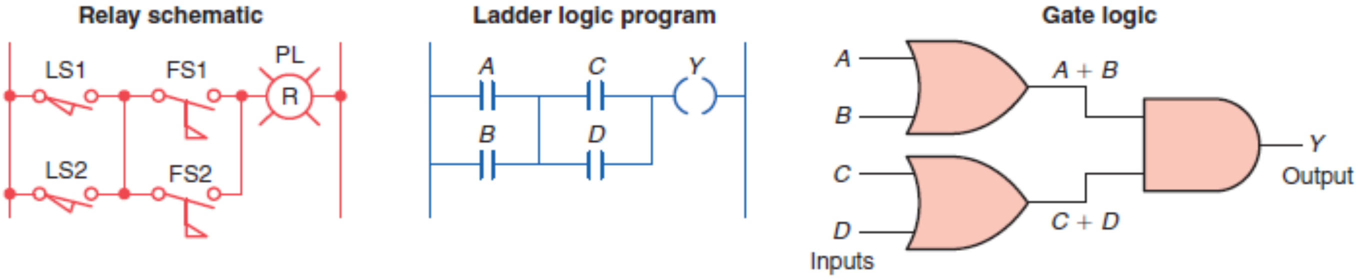


Relay schematic

Ladder logic program

Gate logic

Boolean equation: $(A + B)C = Y$

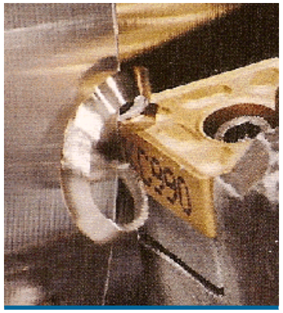


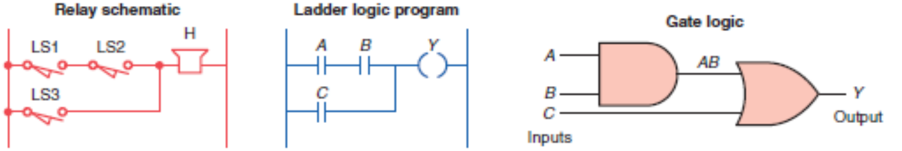
Relay schematic

Ladder logic program

Gate logic

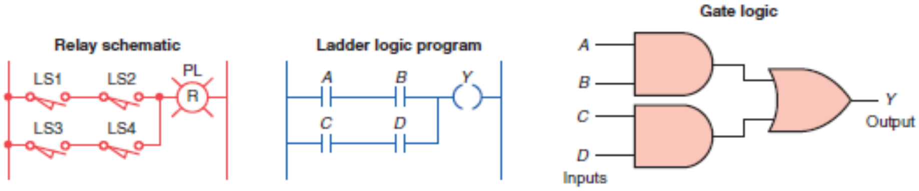
Boolean equation: $(A + B)\ (C + D) = Y$





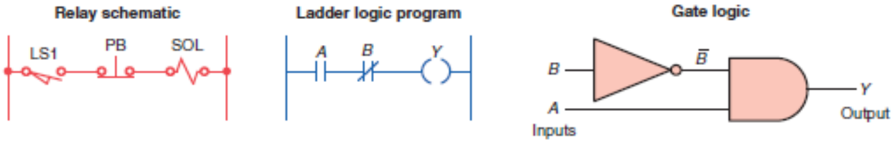
Boolean equation: $(AB) + C - Y$

Example 4-5 Two limit switches connected in series with each other and in parallel with a third limit switch, and used to control a warning horn.



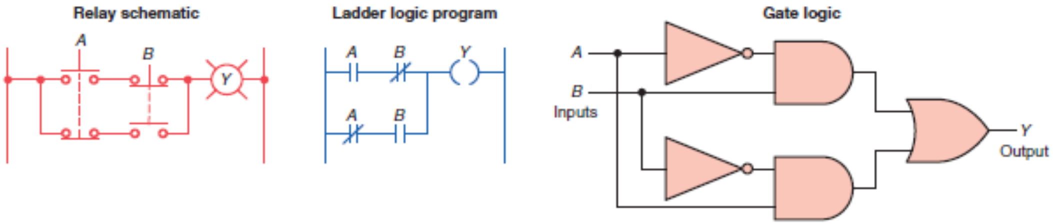
Boolean equation: $(AB) + (CD) - Y$

Example 4-6 Two limit switches connected in series with each other and in parallel with two other limit switches (that are connected in series with each other), and used to control a pilot light.



Boolean equation: $A\overline{B} - Y$

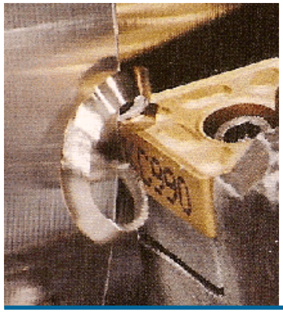
Example 4-7 One limit switch connected in series with a normally closed pushbutton and used to control a solenoid valve. This circuit is programmed so that the output solenoid will be turned on when the limit switch is closed and the pushbutton is *not pushed*.



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

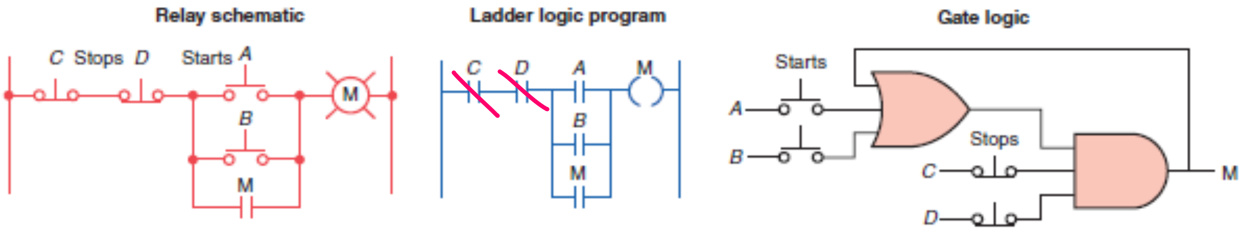
$A \oplus B = Y$

Example 4-8 Exclusive-OR circuit. The output lamp of this circuit is ON only when pushbutton A or B is pressed, but not both. This circuit has been programmed using only the normally open A and B pushbutton contacts as the inputs to the program.



Fundamentals of Logic

دوائر فيها اغلاط بدنا نصلحها



Example 4-9 A motor control circuit with two start/stop buttons. When either start button is depressed, the motor runs. By use of a seal-in contact, it continues to run when the start button is released. Either stop button stops the motor when it is depressed.

* Example :-

$$Z = \bar{A}C\left(\overline{\bar{A}BD}\right) + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}C\left(\overset{A}{\cancel{\bar{\bar{A}}}} + \bar{B} + \bar{D}\right) + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \overset{0}{\cancel{\bar{A}C.A}} + \bar{A}\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}\bar{B}C + A\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D}$$

$$= \bar{B}C\underbrace{(A+\bar{A})}_{1} + \bar{A}\bar{D}\underbrace{(C+B\bar{C})}_{C+B}$$

$$\boxed{= \bar{B}C + \bar{A}\bar{D}(C+B)}$$

$$\overline{A+B} = \bar{A}.\bar{B}$$
$$\overline{A.B} = \bar{A}+\bar{B}$$
$$\bar{\bar{A}} = A$$
$$A.\bar{A} = 0$$
$$A+A = A$$
$$A+\bar{A} = 1$$
$$\bar{A}+AB = \bar{A}+B$$
$$A+\bar{A}B = A+B$$

* Example :-

$$\bar{A}\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}\bar{B}C + \overset{\bar{A}C\bar{D}}{\left(\bar{A}C\bar{D} + \bar{A}C\bar{D}\right)} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}C(\bar{B}+\bar{D}) + (\bar{A}\bar{D}\overset{C+B}{\underline{(C+B\bar{C})}} + A\bar{B}C$$

$$= \bar{A}C(\bar{B}+\bar{D}) + \bar{A}\bar{D}(C+B) + A\bar{B}C$$

$$= \bar{A}\bar{B}C + \underbrace{\bar{A}C\bar{D} + \bar{A}C\bar{D}}_{\bar{A}C\bar{D}} + \bar{A}B\bar{D} + A\bar{B}C$$

$$+ \bar{B}C\underbrace{(A+\bar{A})}_{1} + \bar{A}B\bar{D}$$

$$= \bar{A}C\bar{D} + \bar{B}C + \bar{A}B\bar{D}$$

$$\boxed{= \bar{A}\bar{D}(C+B) + \bar{B}C}$$

Boolean Simplification Algebra

$$A(\bar{B}+A+\bar{B}\bar{C}(A+\bar{B}\bar{C}))$$

$$=A(\bar{B}+A+\underbrace{\bar{B}\bar{C}A}_{}+\underbrace{\bar{B}\bar{C}\bar{B}\bar{C}}_{\bar{B}\bar{C}})$$

عامل مشترك

$$=A(\bar{B}+A+\bar{B}\bar{C}\underbrace{(A+1)}_{1})$$

$$=A(\bar{B}\underbrace{(\bar{C}+1)}_{1}+A)$$

$$=A(\bar{B}+A)$$

$$=A\bar{B}+AA$$

$$=A\underbrace{(\bar{B}+1)}_{1}$$

$$=A$$

$$(A\bar{B}(C+BD)+\bar{A}\bar{B})C$$

$$=(A\bar{B}C+\cancel{A\underbrace{\bar{B}BD}_{\bar{B}B=0}})+\bar{A}\bar{B})C$$

$$CC=C$$

$$=A\bar{B}C+\bar{A}\bar{B}C$$

$$=\bar{B}C\underbrace{(A+\bar{A})}_{1}$$

$$=\bar{B}C$$

$$Z=ABC+A\bar{B}\cdot(\overline{\bar{A}\cdot\bar{C}})$$

$$=ABC+A\bar{B}\cdot(\underbrace{\bar{\bar{A}}}_{A}+\underbrace{\bar{\bar{C}}}_{C})$$

$$=ABC+A\bar{B}A+A\bar{B}C$$

$$=AC(B+\bar{B})+A\bar{B}$$

$$=A(C+\bar{B})$$

$$\bar{X}YZ+(\overline{XY\bar{Z}})$$

$$=\bar{x}yz+(\bar{x}+\bar{y}+\underbrace{\bar{\bar{z}}}_{z})$$

$$=\bar{x}yz+(\bar{x}+\bar{y}+z)$$

عامل مشترك

$$=z\underbrace{(\bar{x}y+1)}_{1}+\bar{x}+\bar{y}$$

$$=\bar{x}+\bar{y}+z$$

$$XY+X(Y+Z)+Y(Y+Z)$$

$$*\quad =XY+XY+XZ+\underbrace{YY}_{Y}+YZ$$

$$=Y\underbrace{(X+1)}_{1}+XZ+YZ$$

$$=Y(1+Z)+XZ$$

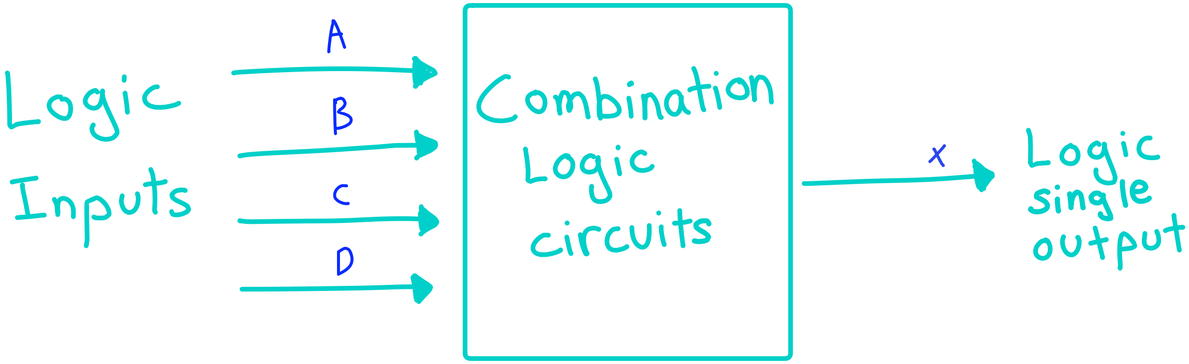
$$=Y+XZ$$

$$Z=A\bar{B}\bar{C}+A\bar{B}C+ABC$$

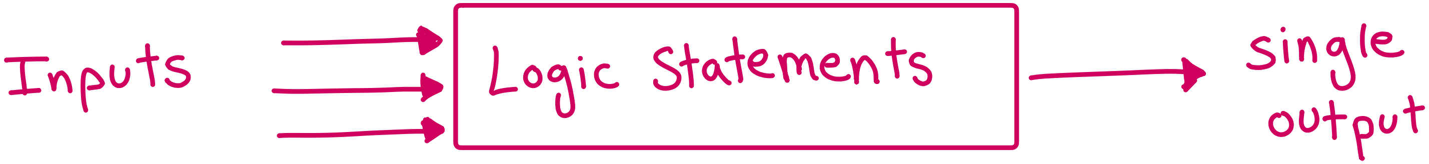
$$=A\bar{B}\bar{C}+AC\underbrace{(\bar{B}+B)}_{1}$$

$$=A(\bar{B}\bar{C}+C)$$

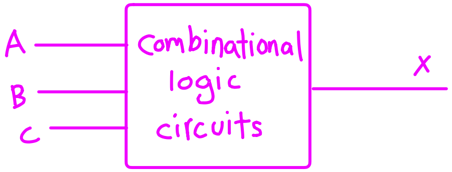
$$=A(C+\bar{B})$$



*based on Logic statments that converts Logic inputs into output



*example



في هذا المثال ↙

Logic statment

[X is high when two or more of the inputs are high]

✱ Combinational Logic Circut Design

1 Truth Table

to connect all combinational possibilities of inputs to output

↳ # of them $= 2^{\text{inputs}} = 2^3 = 8$

	C	B	A	X
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

في هذا المثال ↰

Logic statment

[X is high when two or more of the inputs are high]

2 Write the output as a boolean expression based on the Truth table

$$X = AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC$$

$A + A = A$

$ABC + ABC = ABC$

$A + \bar{A} = 1$

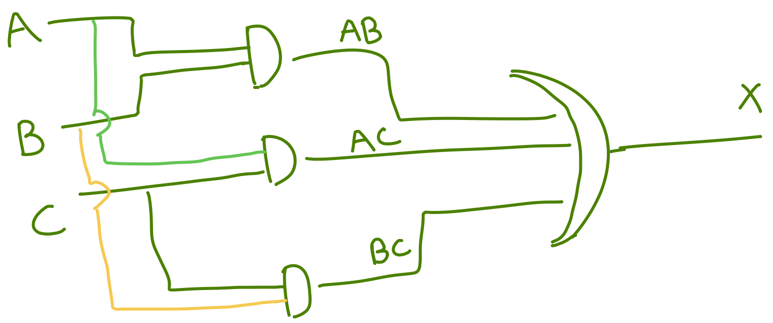
3 Simplify :-

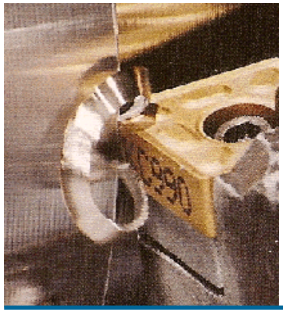
$$X = AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC + ABC + ABC$$

$$= AB(C+\bar{C}) + AC(\bar{B}+B) + BC(A+\bar{A})$$

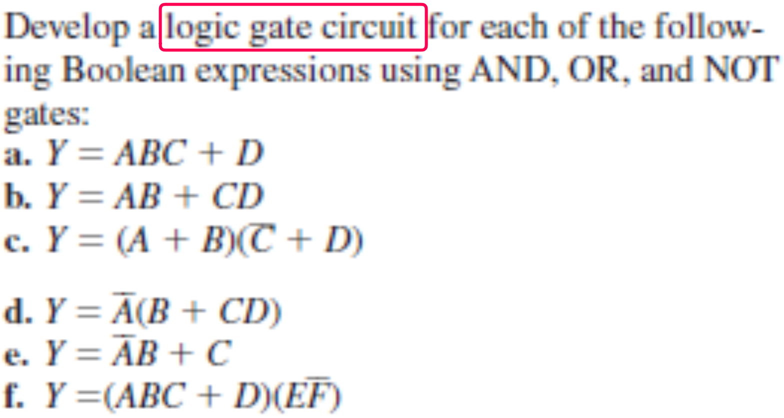
$$= AB + AC + BC$$

4 Draw the Combinational Logic circuit :-





Fundamentals of Logic



Develop a logic gate circuit for each of the following Boolean expressions using AND, OR, and NOT gates:

a. $Y = ABC + D$

b. $Y = AB + CD$

c. $Y = (A + B)(\overline{C} + D)$

d. $Y = \overline{A}(B + CD)$

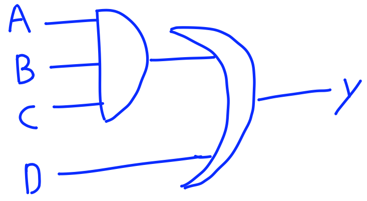
e. $Y = \overline{A}B + C$

f. $Y = (ABC + D)(E\overline{F})$

* Develop a Logic Gate circuit :-

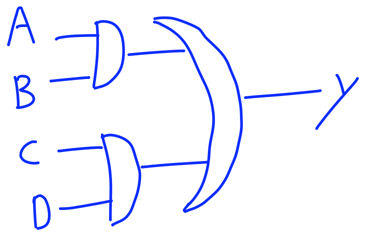
(a)

$Y = ABC + D$



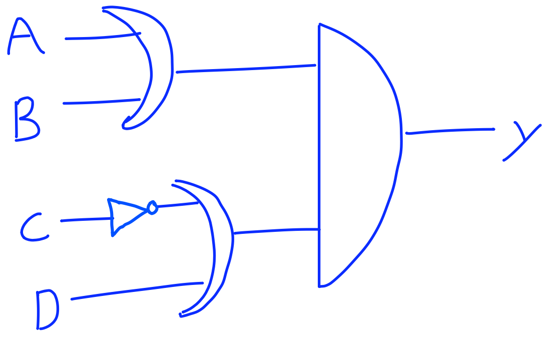
(b)

$Y = AB + CD$



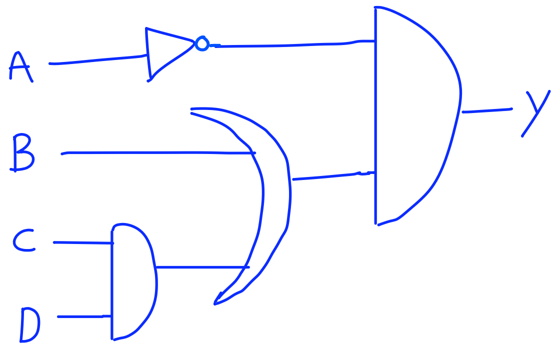
(c)

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



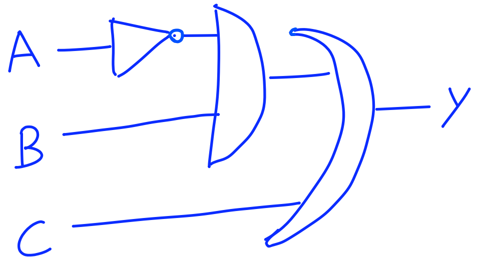
(d)

$Y = \overline{A}(B+CD)$



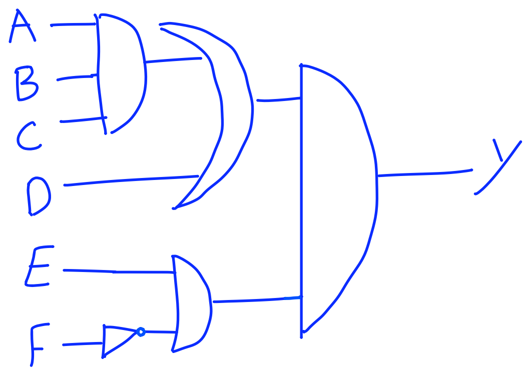
(e)

$Y = \overline{A}B + C$



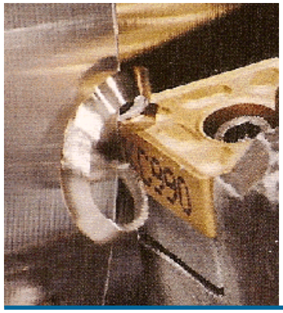
(f)

$Y = (ABC+D)(E\overline{F})$



Ladder diagram :-

* single output
* multiple input
* no feedback

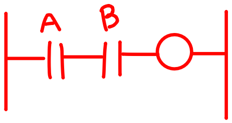


Fundamentals of Logic

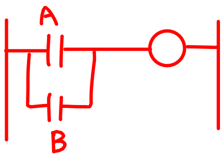
4. Express each of the following equations as a ladder logic program:
   a. $Y = (A + B)CD$
   b. $Y = A\bar{B}C + \bar{D} + E$
   c. $Y = [(\bar{A} + \bar{B})C] + DE$
   d. $Y = (\bar{A}B\bar{C}) + (D\bar{E}F)$

*Gates in Ladder Diagram

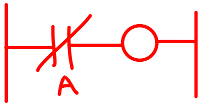
AND Gate
(تَوالي)
(A.B)



OR Gate
(توازي)
(A+B)



Inverter
$\bar{A}$



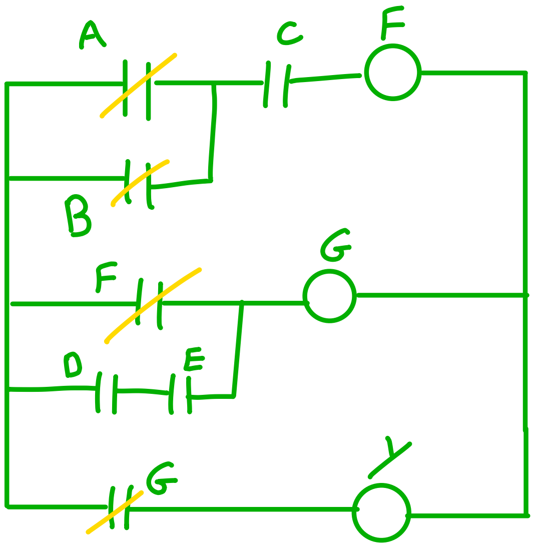
(Always start from inside the brackets)

* Express each of the following equation as a Ladder

$$Y = \overline{\underbrace{\overline{\underbrace{(\bar{A}+\bar{B}).C}_{F}}}_{\bar{F}} + DE}_{G}$$

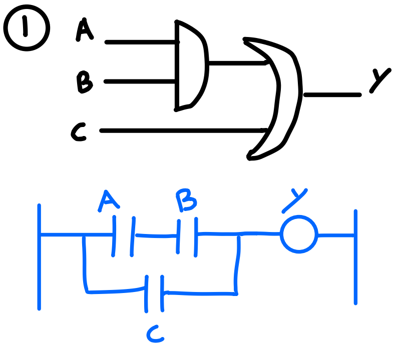
$$\boxed{Y = \overline{\overline{(\bar{A}+\bar{B}).C} + DE}}$$

يبدأ دائماً بالجزء الداخل الاقواس

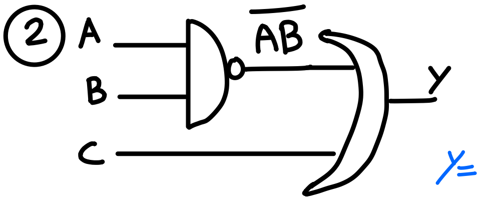


* Convert from (combinational Logic circuit) to (Ladder Diagram)

①



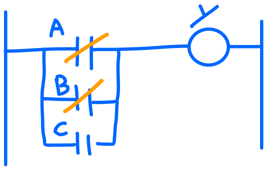
②



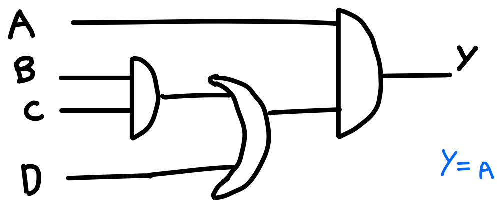
$Y = \overline{AB} + C$

↳ Demorgan's Theory

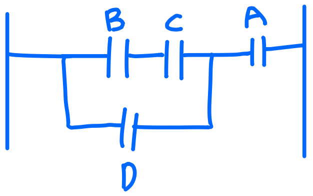
$\boxed{\bar{A} + \bar{B} + C}$



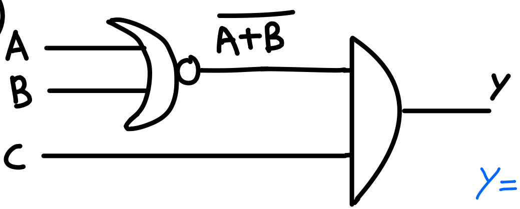
③



$Y = A(BC+D)$

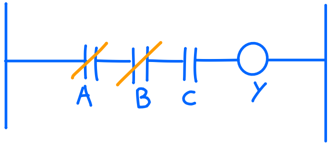


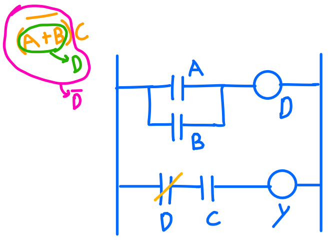
④



$Y = (\overline{A+B})C$

$= \bar{A} \cdot \bar{B} \cdot C$





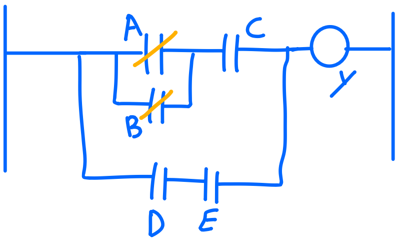
أو

*Note :

D and Y are outputs so we should draw them above each others

⑤

$Y = [(\bar{A}+\bar{B})C] + DE$

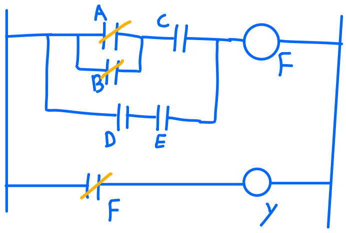


⑥

$Y = \overline{[(\bar{A}+\bar{B})C] + DE}$

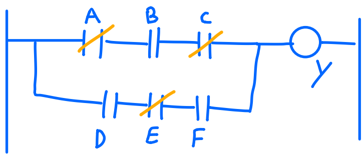
→ F

→ $\bar{F}$

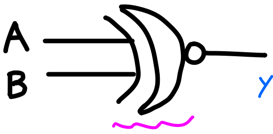


⑦

$Y = (\bar{A}B\bar{C}) + (D\bar{E}F)$

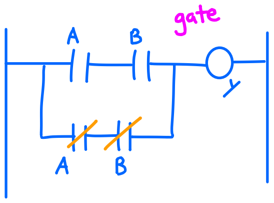


⑧

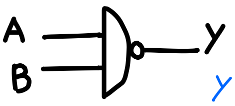


$Y = AB + \bar{A}\bar{B}$

XNOR gate

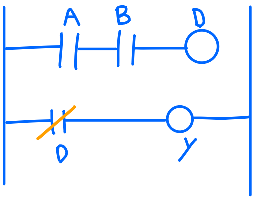


⑨



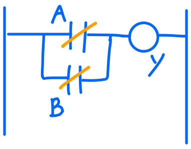
$Y = \overline{AB}$

NAND gate

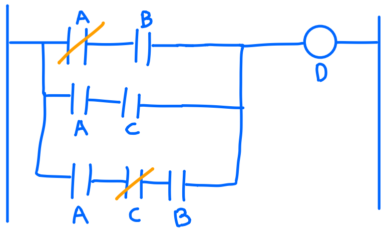


أو with simplification

$Y = \bar{A} + \bar{B}$



⑩ $D = \bar{A}.B + A.C + A.\bar{C}.B$



أو
with simplification

$$D = \bar{A}.B + \underbrace{A.C + A.\bar{C}.B}_{\text{عامل مشترك}}$$

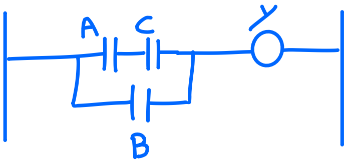
$$= \bar{A}B + A(C+\bar{C}B)$$

$$= \bar{A}B + A(C+B)$$

$$= \underbrace{\bar{A}B} + AC + \underbrace{AB}$$

$$= B(\bar{A}+A) + AC$$

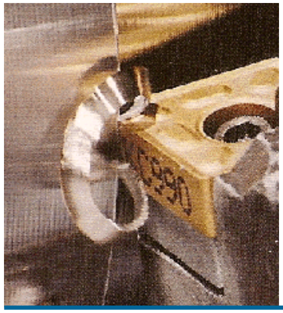
$$= B + AC$$



Note:-

يا برسم ال Ladder diagram من البداية

يا بسوي simplifications بعدين برسمه

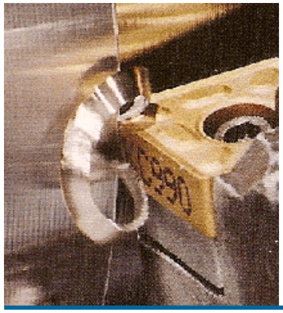


Example 1:

- On a particular piece of operator-controlled production equipment, the production process may only be performed by the operator activating two safety switches, located at some distance from each other. This is to prevent the equipment from accidentally starting whilst the operator is loading or unloading the machine. The switches have to be depressed together by the operator using both hands.

That means (AND gate)

- (a) What is the truth table for this operation?
- (b) What is the Boolean logic expression for this operation?
- (c) What is the logic network diagram for the operation?

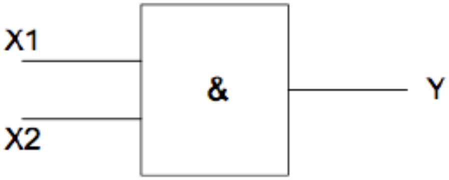


Example 1 solution:

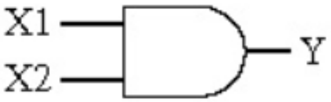
- (a) Where X1 is first switch, and X2 is second switch, and Y is the output of switch activation.

Inputs		Output
X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

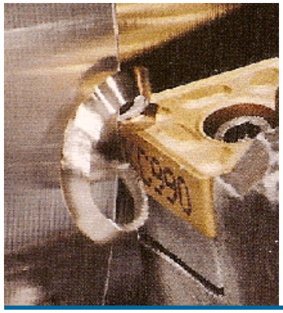
- (b) Y = X1*X2
- (c)



logic network diagram

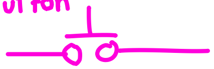


✓

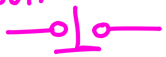


Example 2:

No push button



NC push button

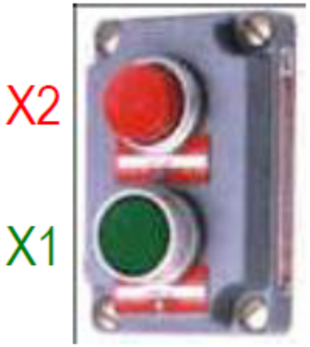


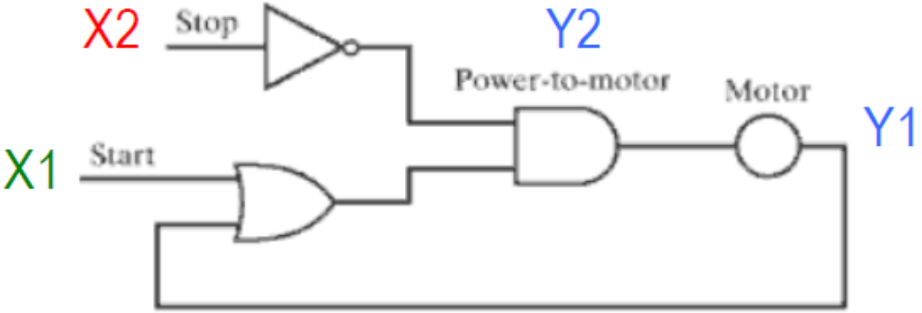
Write the Boolean logic expression for the pushbutton switch system below using the following symbols:

X1 = START, X2 = STOP, Y1 = MOTOR, and Y2 = POWER-TO-MOTOR.

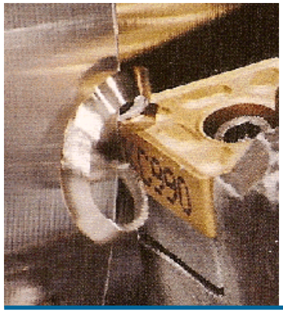
بتطفي ←
الsystem
بس ما بتشغله

بتشغل ←
الsystem
بس ما بتطفيه

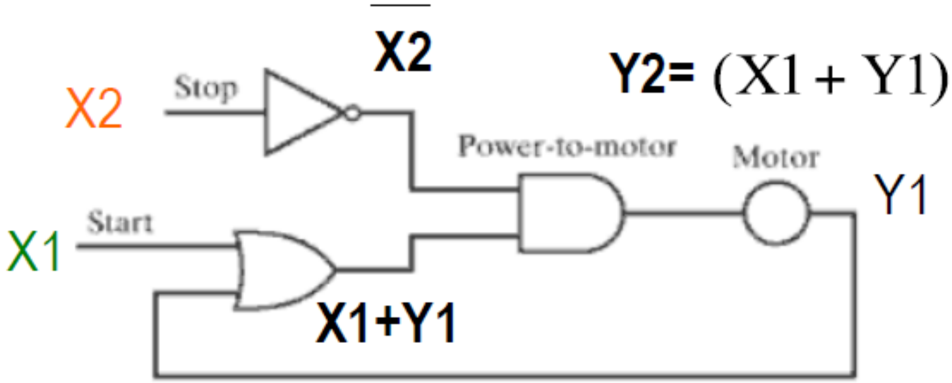




logic network diagram



Example 2 Solution:



$$Y2 = (X1 + Y1) \cdot \overline{X2}$$

Truth Table

$= \overline{X_2}(X_1+Y_1)$

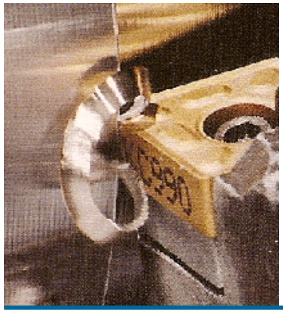
	Start	Stop	Motor	Power-to-Motor
	0	0	0	0
	0	1	0	0
	1	0	0	1
x	1	1	0	0
	0	0	1	1
	0	1	1	0
	1	0	1	1
x	1	1	1	0

This Truth table :-

→ Logically is correct

→ but in Reality it is not correct

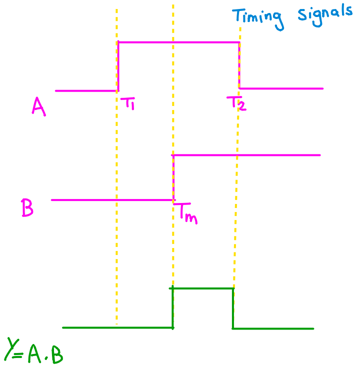
the Automation is sequencing Logic events

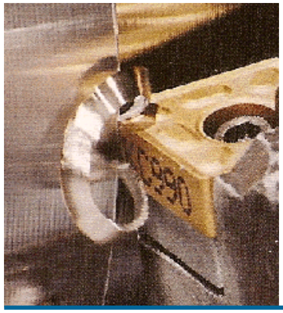


Sequencing

A switching system that uses internal timing devices to determine when to initiate changes in output variables

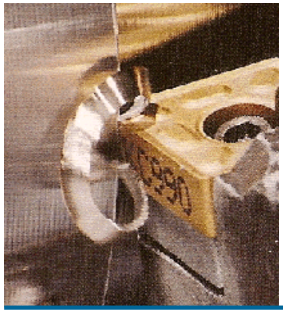
- Examples: Washing machines, dryers, dishwashers, Traffic light.





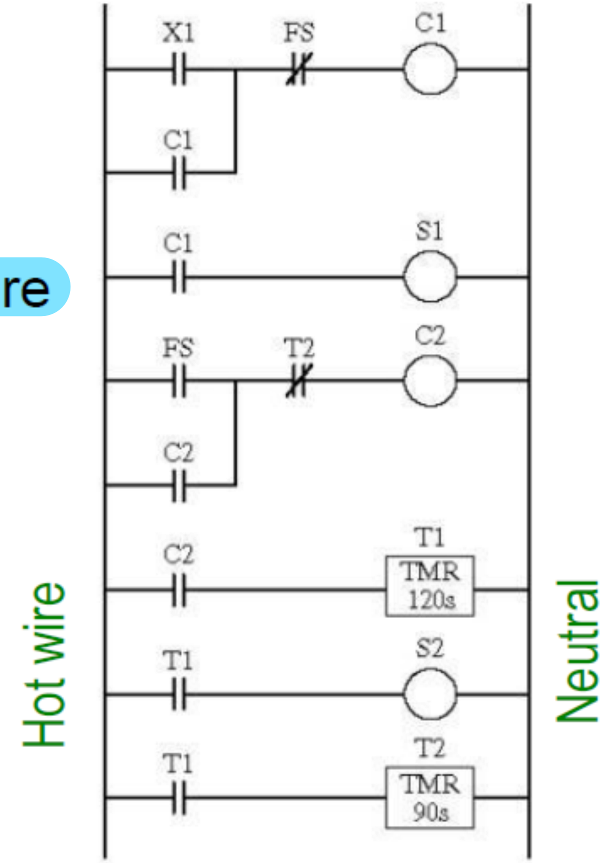
Sequencing

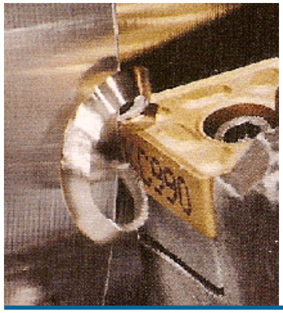
- Outputs are usually generated "open loop"
  - No feedback that control function is executed
- Sequence of output signals is usually cyclical, as in a high production work cycle
  - The signals occur in the same repeated pattern within each regular cycle
- Common sequencing devices:
  - Timer – output switches on/off at preset times
  - Counter – counts electrical pulses and stores them



Ladder Logic Diagrams

- Another way for drawing Logic Network Diagrams.
- A diagram where logic elements are displayed along horizontal lines (rungs) connecting two rails.
- Combines both: logic and sequencing control.





Ladder Logic Diagrams

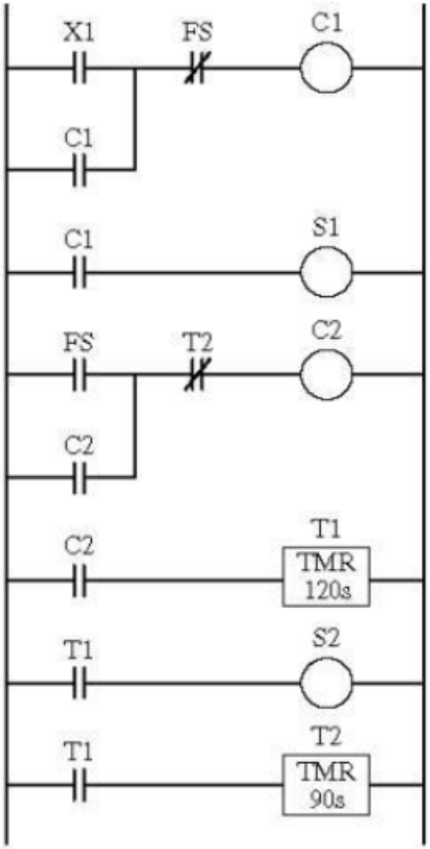
- Components of Ladder Diagrams:

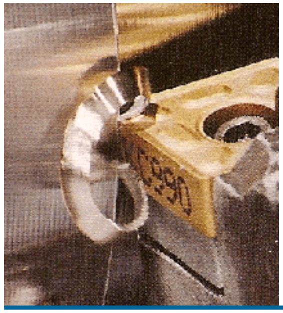
1. Contacts - logical inputs, e.g., limit switches, photo-detector.

2. Loads (coils) - outputs, e.g., motors, lights, alarms, solenoids.

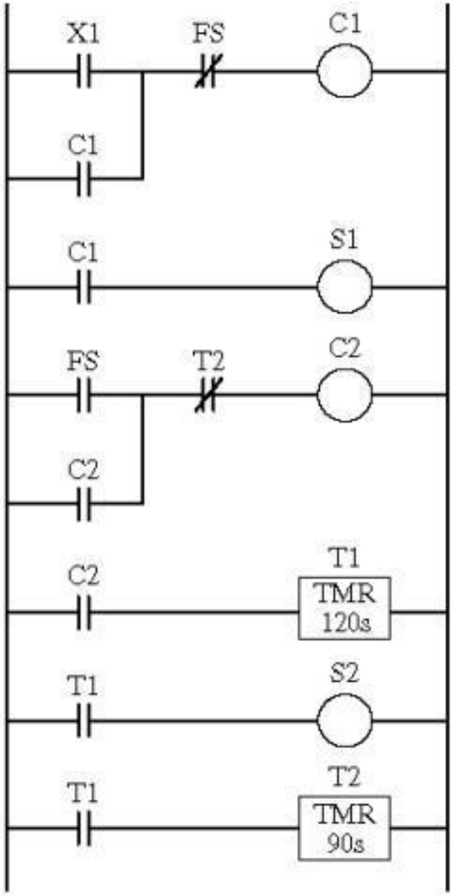
3. Timers - to specify length of delay.

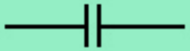
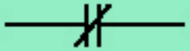
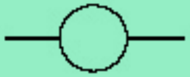
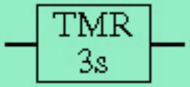
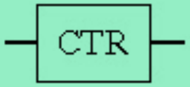
4. Counters - to count pulses received

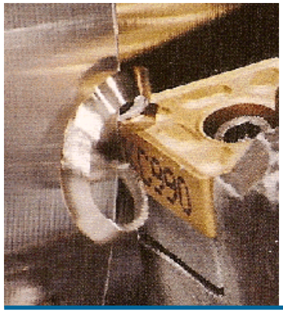




Components of Ladder Logic Diagram



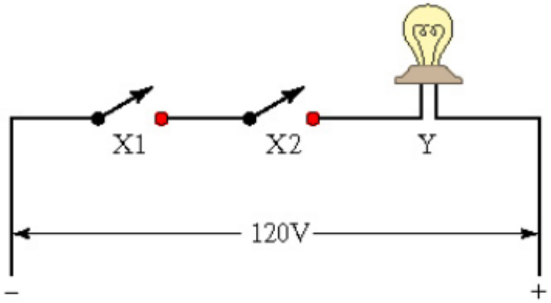
Ladder symbol	Hardware component
(a) 	Normally open contacts (switch, relay, other ON/OFF devices)
(b) 	Normally closed contacts (switch, relay, etc.)
(c) 	Output loads (motor, lamp, solenoid, alarm, etc.)
(d)  TMR 3s	Timer
(e)  CTR	Counter

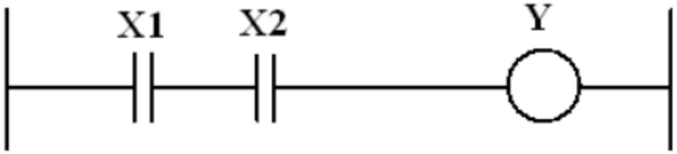


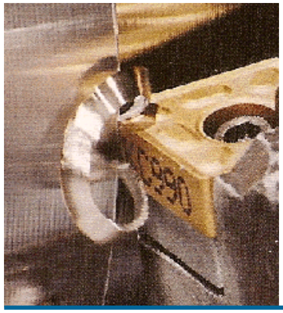
Example 1

- Construct the ladder logic diagrams for the AND gate.
  (series)

- Solution:





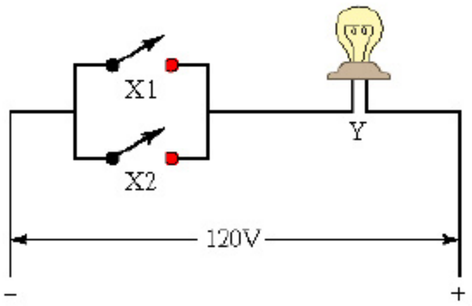


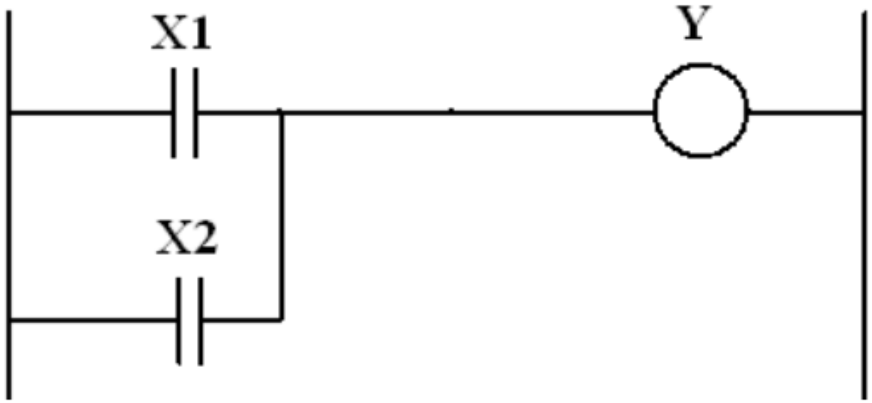
Example 2

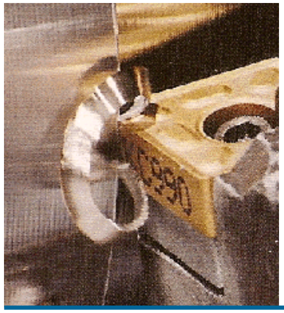
- Construct the ladder logic diagrams for the OR gate.

(parrallel)

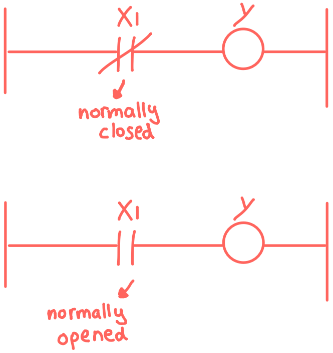
- Solution:



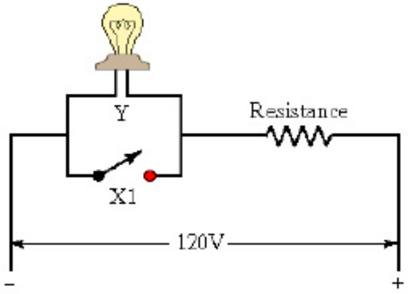


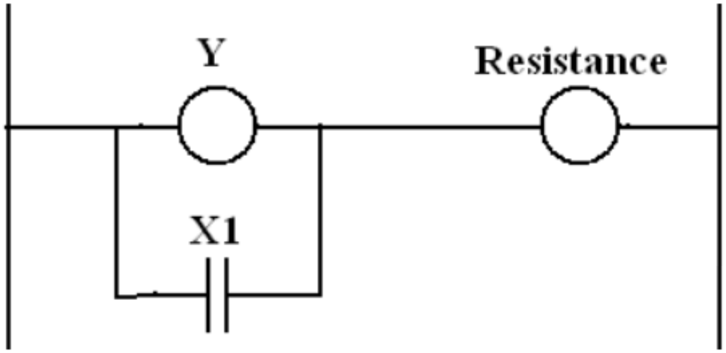


Example 3



- Construct the ladder logic diagrams for the NOT gate.
- Solution:

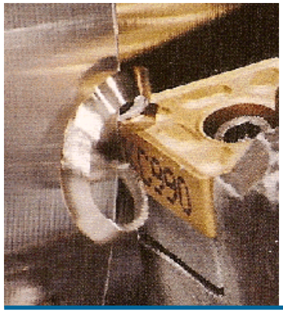




التيار سوف يمر ب X1 لما نسكر ال switch لأنها short circuit ف ال LED سوف ينطفيء

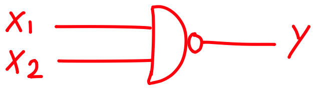
$X_1 : 0 \rightarrow Y : 1$

$X_1 : 1 \rightarrow Y : 0$

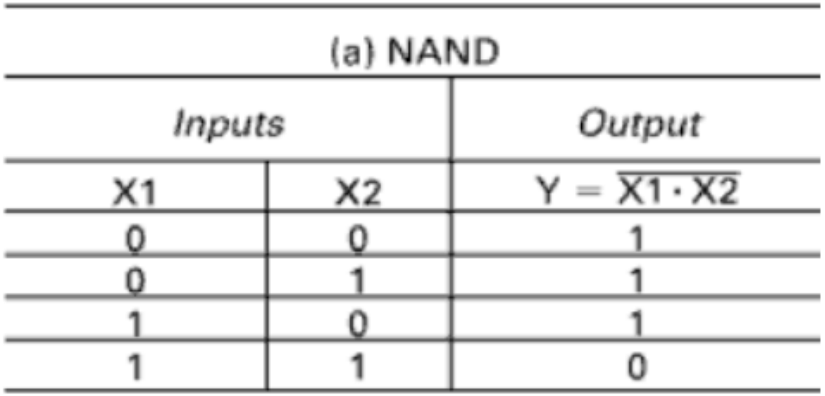


Example 4

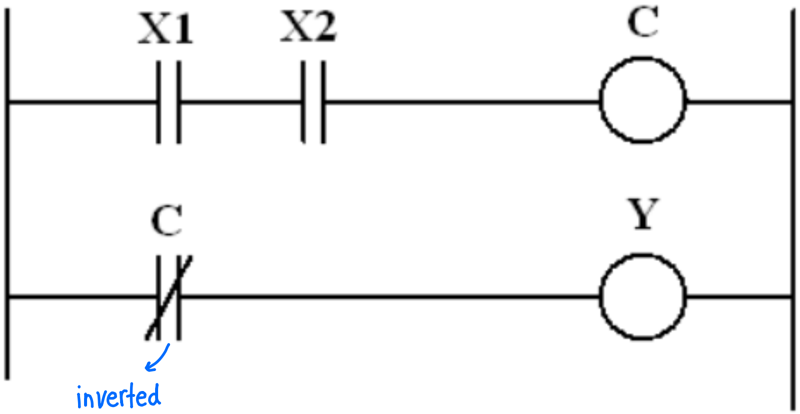
- Construct the ladder logic diagrams for the NAND gate.
- Solution:



$Y = \overline{X_1 . X_2} \longrightarrow$ we can simplify $\overline{X_1} + \overline{X_2}$

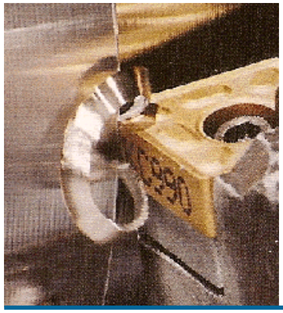


(a) NAND		
Inputs		Output
X1	X2	$Y = \overline{X1 \cdot X2}$
0	0	1
0	1	1
1	0	1
1	1	0



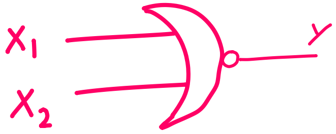
- If X1 or X2 remain open then C coil is unexcited and C contact remains closed, therefore Y is on.
- If X1 and X2 are closed then C coil is excited and C contact is opened and Y is off

C عكس Y



Example 5

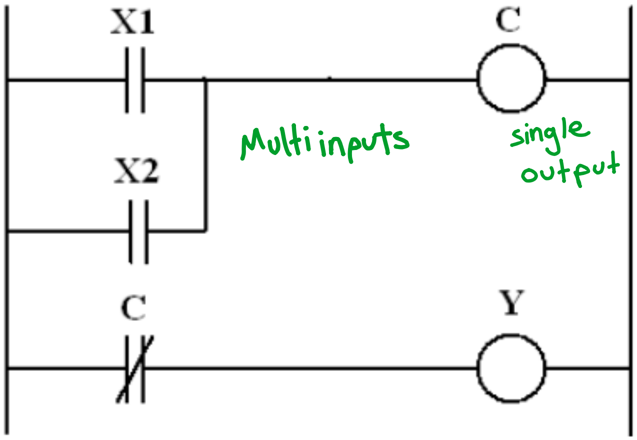
- Construct the ladder logic diagrams for the NOR gate.
- Solution:



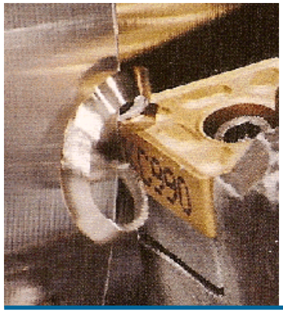
$$Y = \overline{X_1 + X_2} \xrightarrow{\text{we can simplify}} \overline{X_1} \cdot \overline{X_2}$$

(b) NOR

Inputs		Output
X1	X2	$Y = \overline{X1 + X2}$
0	0	1
0	1	0
1	0	0
1	1	0



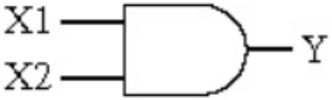
- If X1 and X2 remain open then C coil is unexcited and C contact remains closed, therefore Y is on.
- If X1 or X2 are closed then C coil is excited and C contact is opened and Y is off

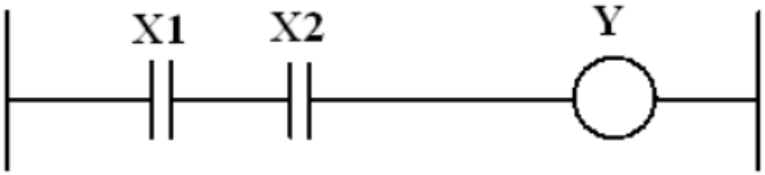


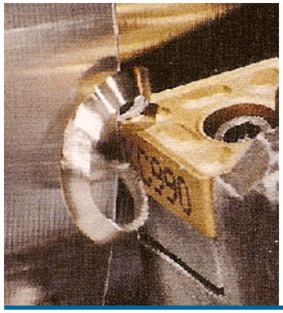
Example 6: Safety switches

- The production process may only be performed when the operator activates two spring activated safety switches. The switches have to be depressed and held closed together by the operator using both hands.

AND



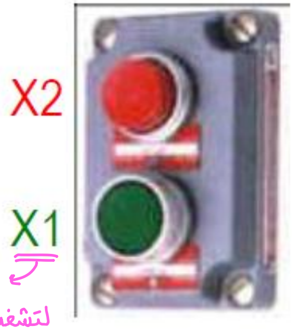




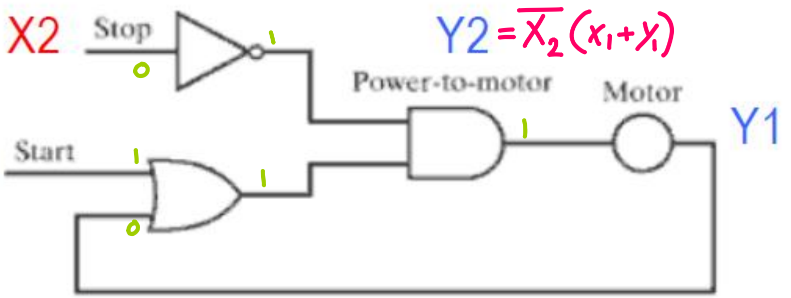
Example 7: Push Button

- Create ladder logic diagram for Push Button switch.

مهم



فقط لتشغيل الـ motor

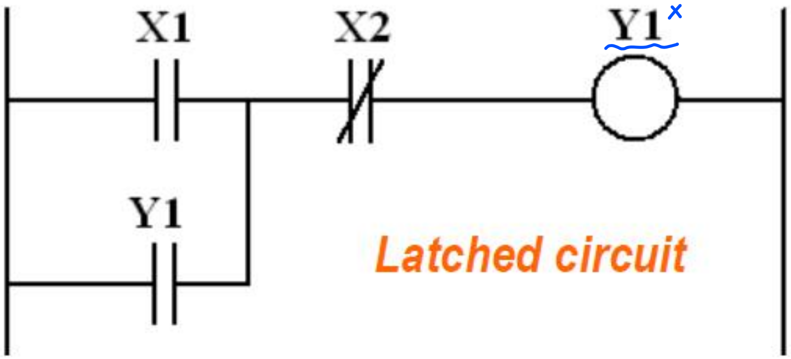


$Y2 = \overline{X_2}(x_1 + Y_1)$

(X1 OR Y1) AND (NOT X2)

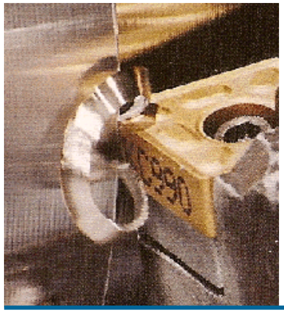
$$(X1 + Y1) \cdot \overline{X2}$$

Y2 اظن



Latched circuit

* لا يوجد Feedback في الـ Ladder Diagram



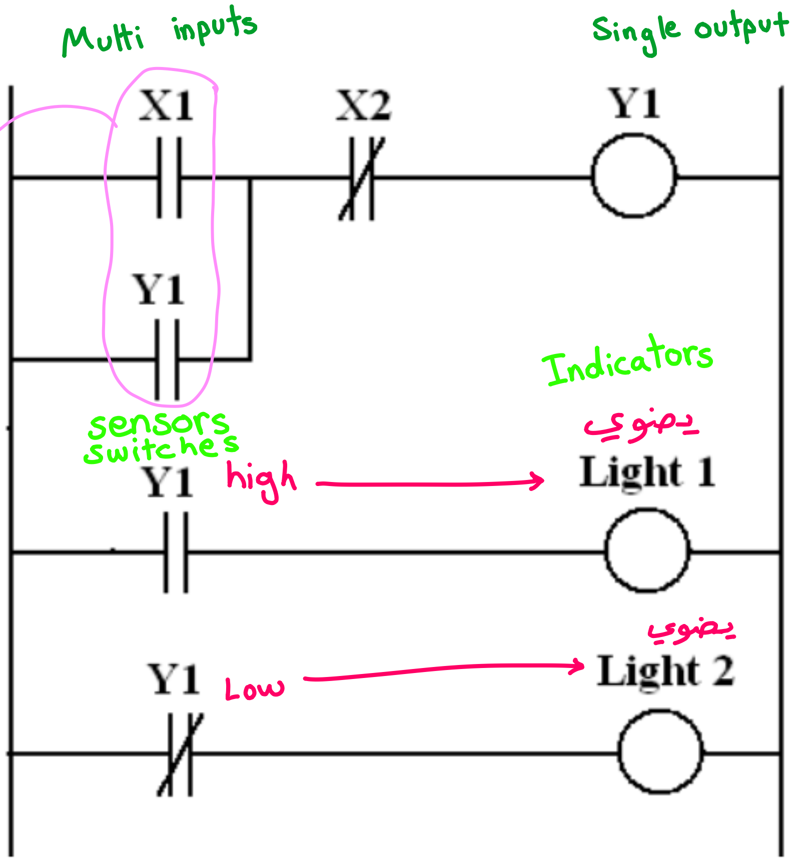
Example 8

- A motor controlled by stop and start push button switches.
- One signal light must be illuminated when the power is applied to the motor and another when it is not applied.

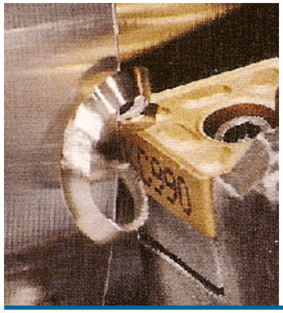
X1 oring with Y1

عشان بس اكبس X1
اكتر من مرة يضل شغال
طبيعي و بس اكبس X2 يسكر

X1 → ON only
X2 → OFF only



كل Rung عبارة عن one event



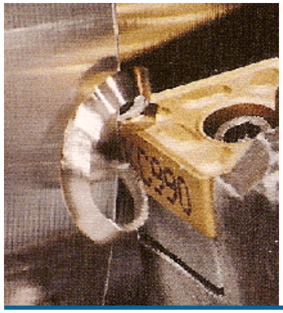
C

Example 9.5 Control Relay

electro magnitic device

يتحكم بـ

- A control relay can be used to control on/off actuation of a powered device at some remote location. It can also be used to define alternative decisions in logic control. Construct the ladder logic diagram of a control relay.

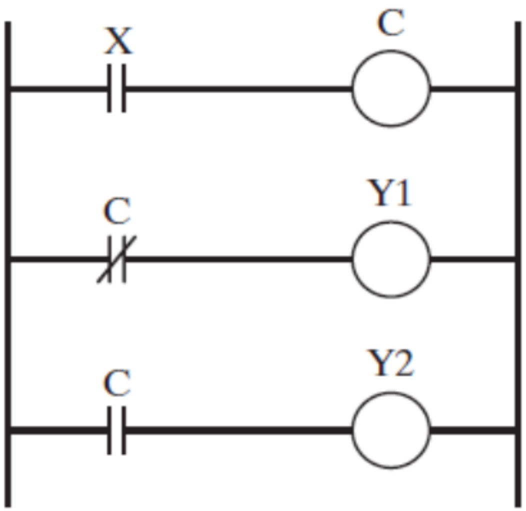


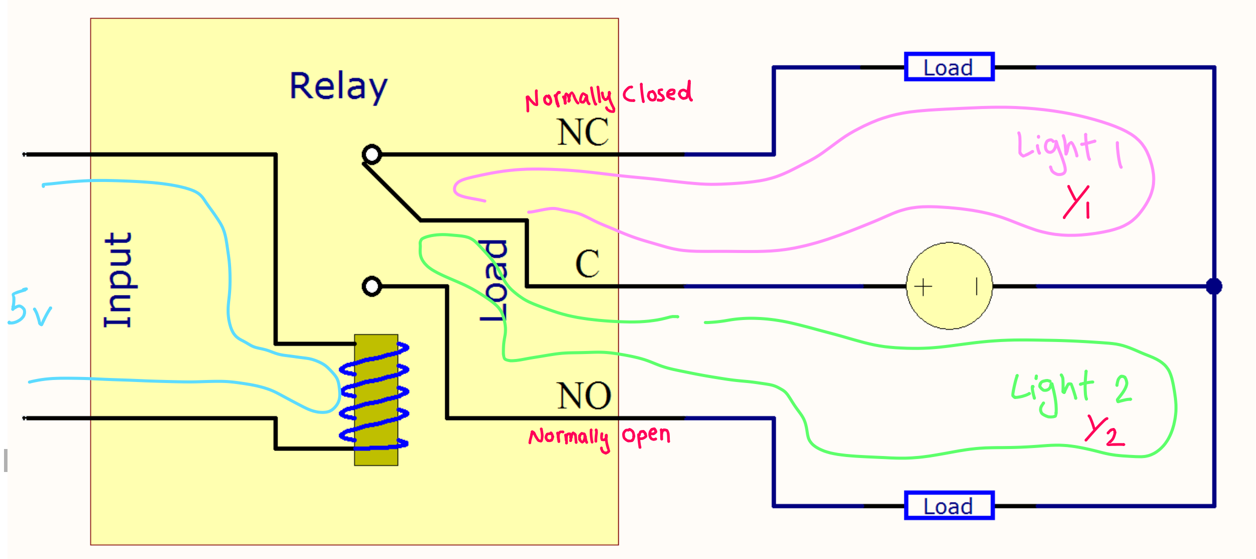
Example 9.5 Control Relay

important

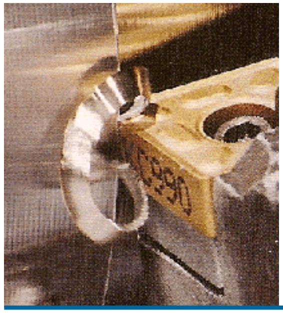
- Electromagnetic relay is an electronic control device. It has a control system (also called an input loop) and a controlled system (also called an output loop). It is usually used in automatic control circuits. It actually a kind of "automatic switch" that uses a smaller current and a lower current to control a larger current and a higher voltage.

X_1	C_1	Y_1	Y_2
1	1	0	1
0	0	1	0





في البداية نحدد ال inputs وال outputs



Example 9.6 Fluid Storage Tank

هدف هذا المثال دراسة ال dependencies between inputs and outputs

- Consider the fluid storage tank illustrated in Figure 9.10.
- When the start button X1 is depressed, this closes the control relay C1, which energizes solenoid S1, which opens a valve allowing fluid to flow into the tank.
- When the tank becomes full, the float switch FS closes, which opens relay C1, causing the solenoid S1 to be de-energized, thus turning off the in-flow.
- Switch FS also activates timer T1, which provides a 120-sec delay for a certain chemical reaction to occur in the tank.

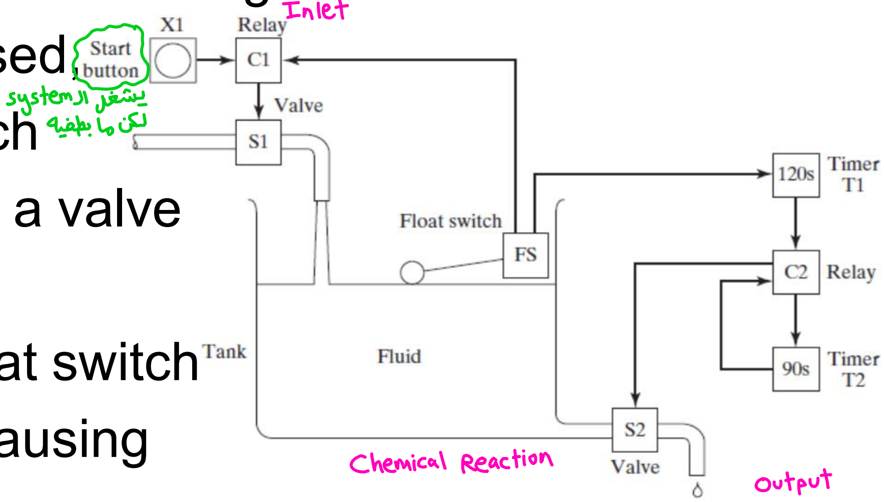
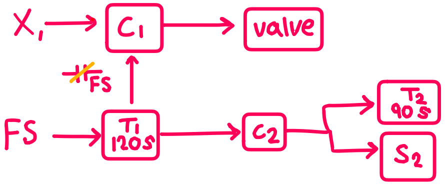
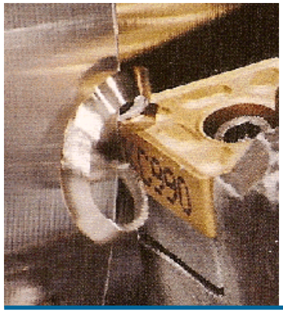


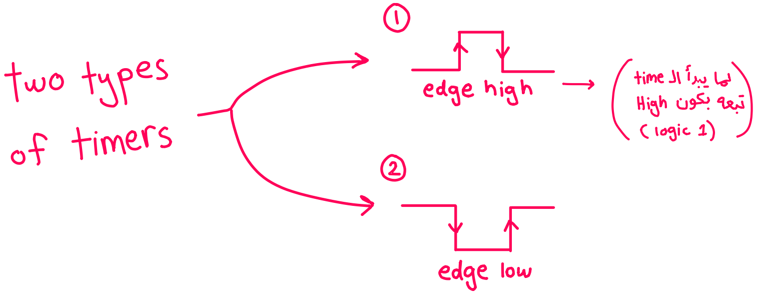
Figure 9.10 Fluid filling operation of Example 9.6.



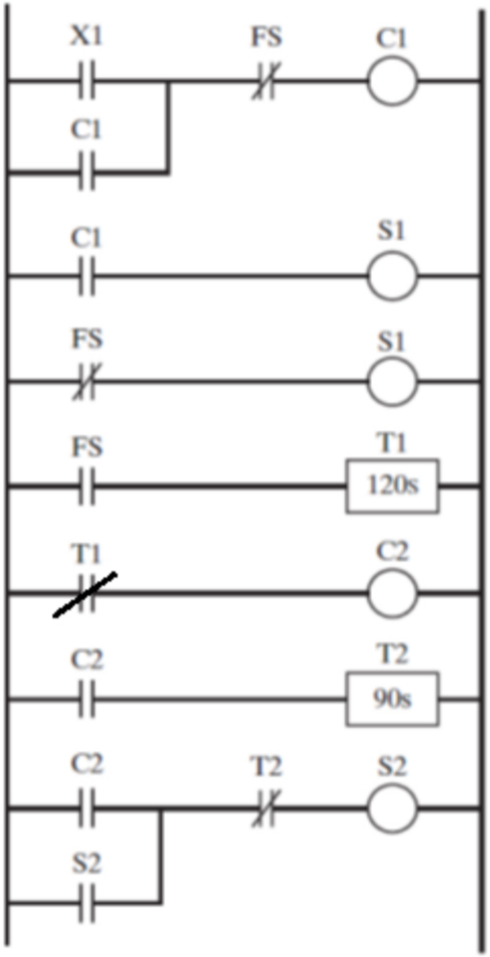


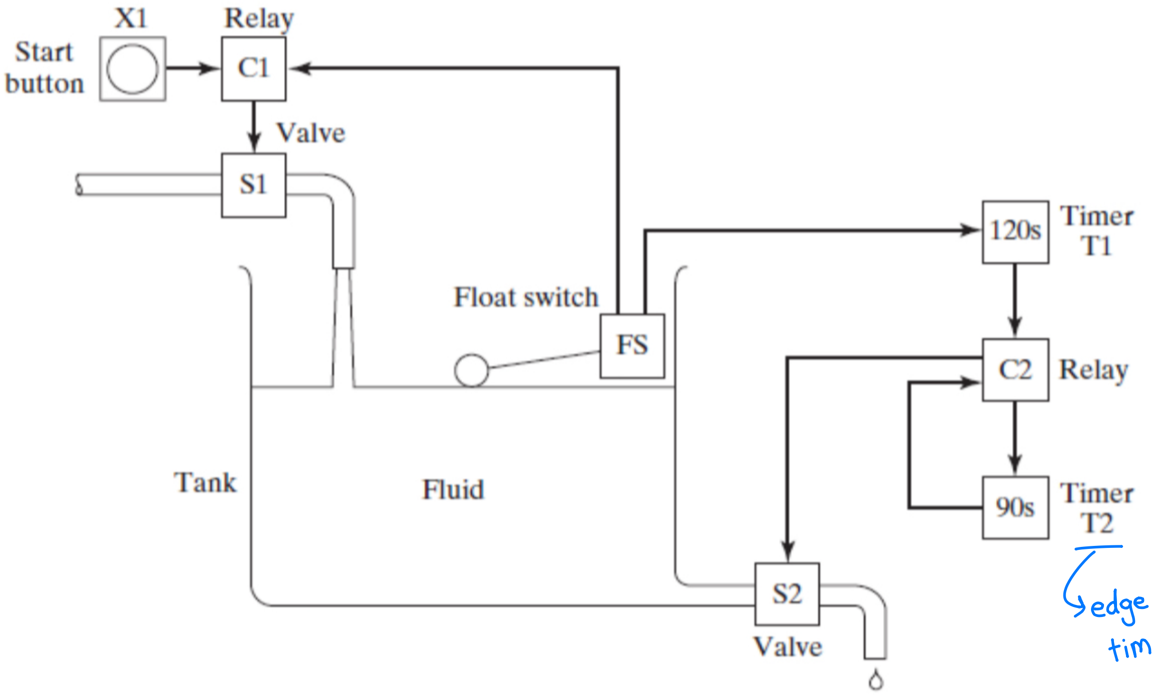
Example 9.6 Fluid Storage Tank

- At the end of the delay time, the timer energizes a second relay C2, which controls two devices: (1) It initiates timer T2, which
- waits 90 sec to allow the contents of the tank to be drained, and (2) it energizes solenoid S2, which opens a valve to allow the fluid to flow out of the tank.
- At the end of the 90 sec, the timer breaks the current and de-energizes solenoid S2, thus closing the out-flow valve.
- Depressing the start button X1 resets the timers and opens their respective contacts.



Example 9.6 Fluid Storage Tank

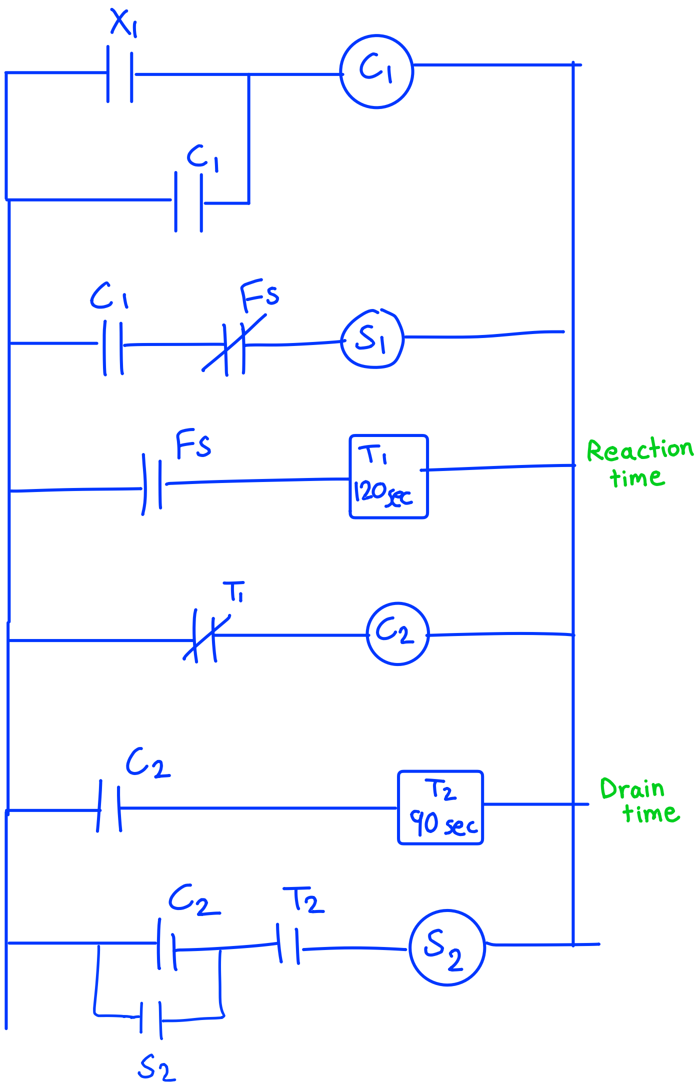


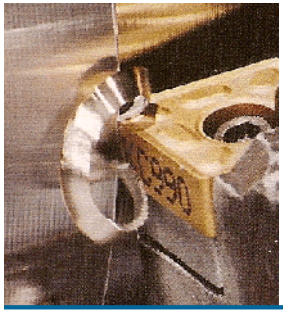


edge low timer

Figure 9.10 Fluid filling operation of Example 9.6.

Timer will be logic 1 when it's time is expired.


X1
C1
C1
C1
Fs
S1
Fs
T1
120sec
Reaction time
T1
C2
C2
T2
90 sec
Drain time
C2
T2
S2
S2



Question

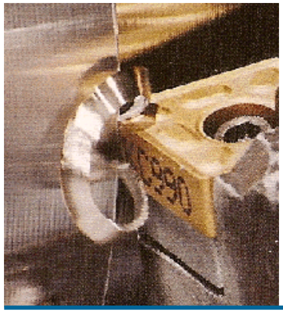
- An industrial robot performs a machine loading and unloading operation. A PLC is used as the cell controller. The cell operates as follows:

(1) a human worker places a part into a nest,

(2) the robot reaches over and picks up the part and places it into an induction heating coil,

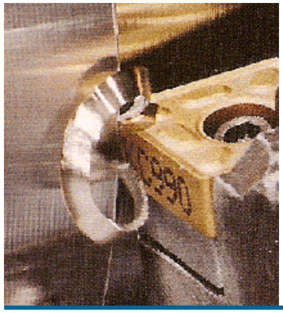
(3) a time of 10 sec is allowed for the heating operation, and (

4) the robot reaches into the coil, retrieves the part, and places it on an outgoing conveyor.



Question

- A limit switch X1 (normally open) is used to indicate that the part is in the nest in step (1). This energizes output contact Y1 to signal the robot to execute step (2) of the work cycle (this is an output contact for
- the PLC, but an input interlock signal for the robot controller).
- A photocell X2 is used to indicate that the part has been placed into the induction heating coil C1.
- Timer T1 is used to provide the 10-sec heating cycle in step (3), at the end of which, output contact Y2 is used to signal the robot to execute step (4).
- Construct the ladder logic diagram for the system



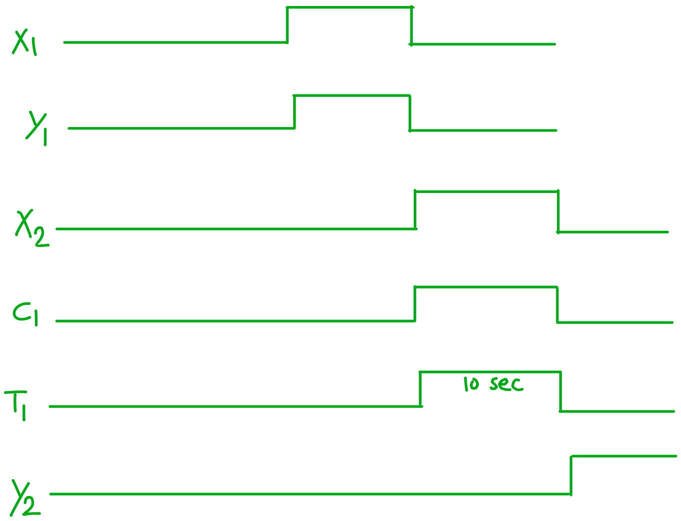
Answer :

→ c_1 is on if the timer is on and there is a part on heating coil

inputs → x_1, → x_2

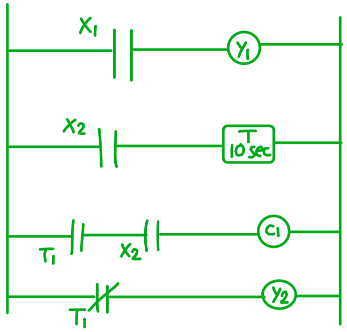
Timers → T_1

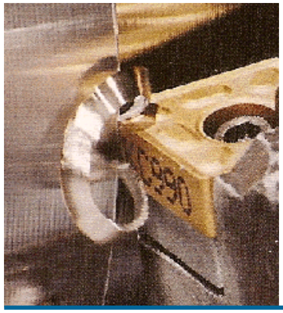
outputs → Y_1 robot signal nest → heating coil
→ c_1 heating coil
→ Y_2 robot signal



⟹

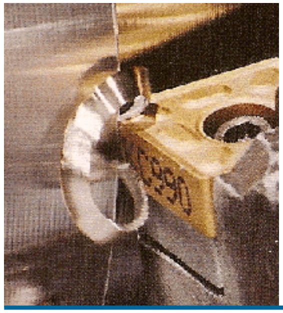
Ladder diagram





Another Question:

- An emergency stop system is to be designed for a certain automatic production machine.
- A single “start” button is used to turn on the power to the machine at the beginning of the day.
- In addition, there are two “stop” buttons located at two locations on the machine, either of which can be pressed to immediately turn off power to the machine.
- Let X1 = start buttom (normally open), X2 = stop button 1 (normally closed), X3 = stop button 2 (normally
- closed), and Y = power to the machine.

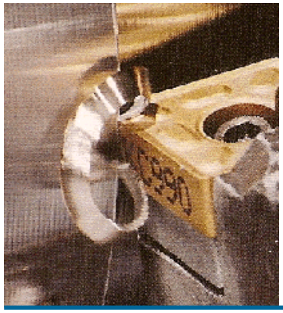


Another Question:

(a) Construct the truth table for this system.

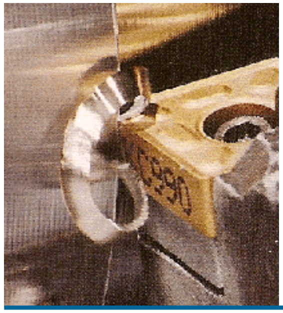
(b) (b) Write the Boolean logic expression for the system.

(c) (c) Construct the ladder logic diagram for the system.



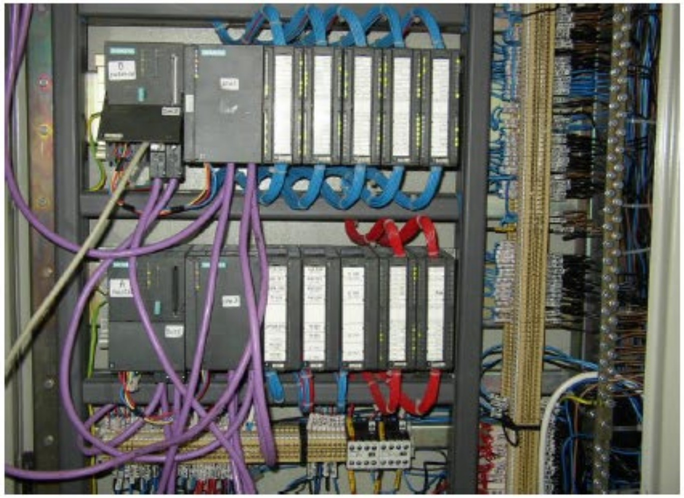
Sec 9.3: Programmable Logic Controller (PLC)

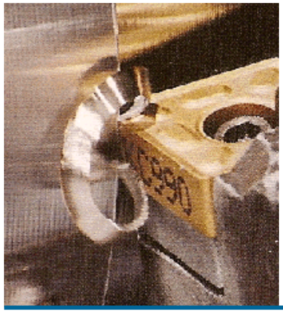
- https://www.youtube.com/watch?v=PbAGI_mv5XI
- https://www.youtube.com/watch?v=pPUnihpL6UI
- https://www.youtube.com/watch?v=wICG8d2iQ5c



Sec 9.3: Programmable Logic Controller (PLC)

- A microcomputer-based controller that uses stored instructions in programmable memory to implement logic, sequencing, timing, counting, and arithmetic functions through digital or analog modules, for controlling machines and processes.





Components of a PLC

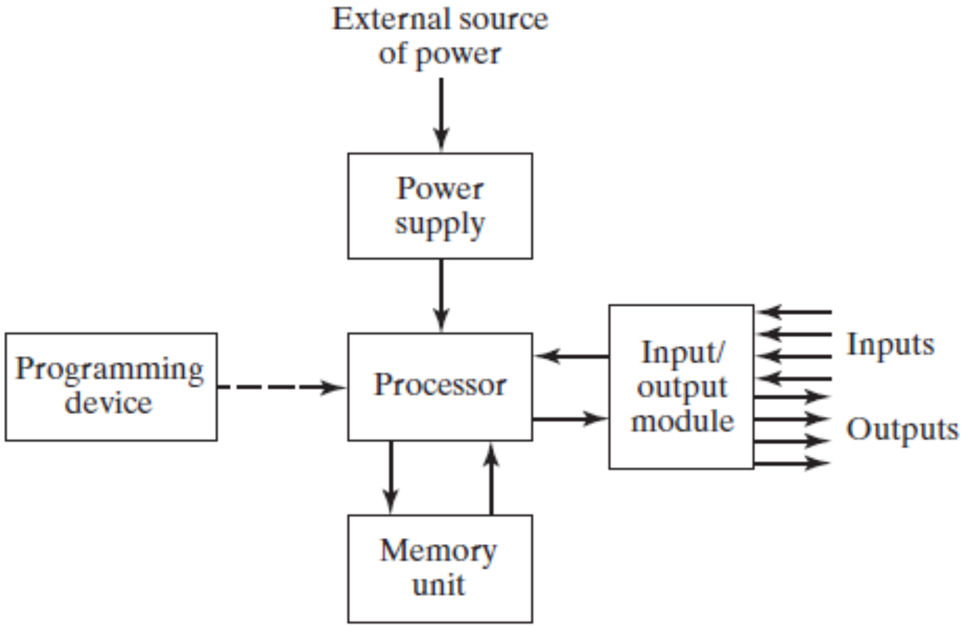
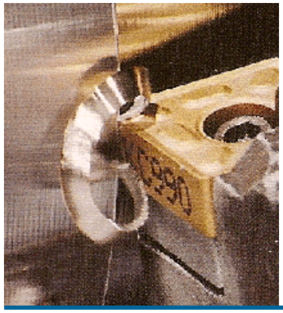
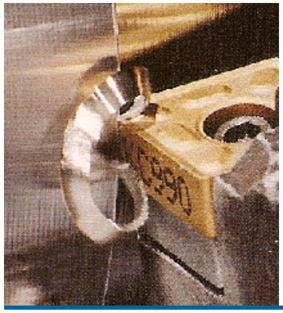


Figure 9.11 Components of a PLC.



Advantages of PLCs Compared to Relay Control Panels

- Programming a PLC is easier than wiring a relay control
- panel
- PLC can be reprogrammed
- PLCs take less floor space
- Greater reliability, easier maintenance
- PLC can be connected to computer systems (CIM)
- PLCs can perform a greater variety of control functions



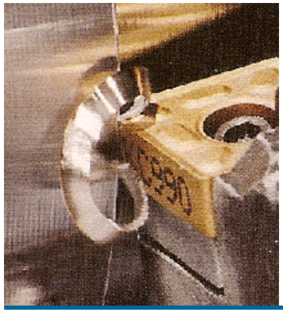
Typical PLC Operating Cycle

1. Input scan – inputs are read by processor and stored in memory
2. Program scan – control program is executed

- Input values stored in memory are used in the control
- logic calculations to determine values of outputs

3. Output scan – output values are updated to agree with

- calculated values
- Time to perform the three steps (scan time) varies between 1 and 25 msec



PLC Programming

- Graphical languages:

  1. Ladder logic diagrams – most widely used

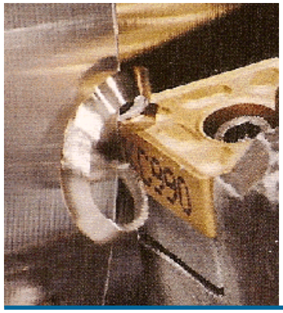
  2. Function block diagrams – instructions composed of operation blocks that transform input signals

  3. Sequential function charts – series of steps and transitions from one state to the next (Europe)

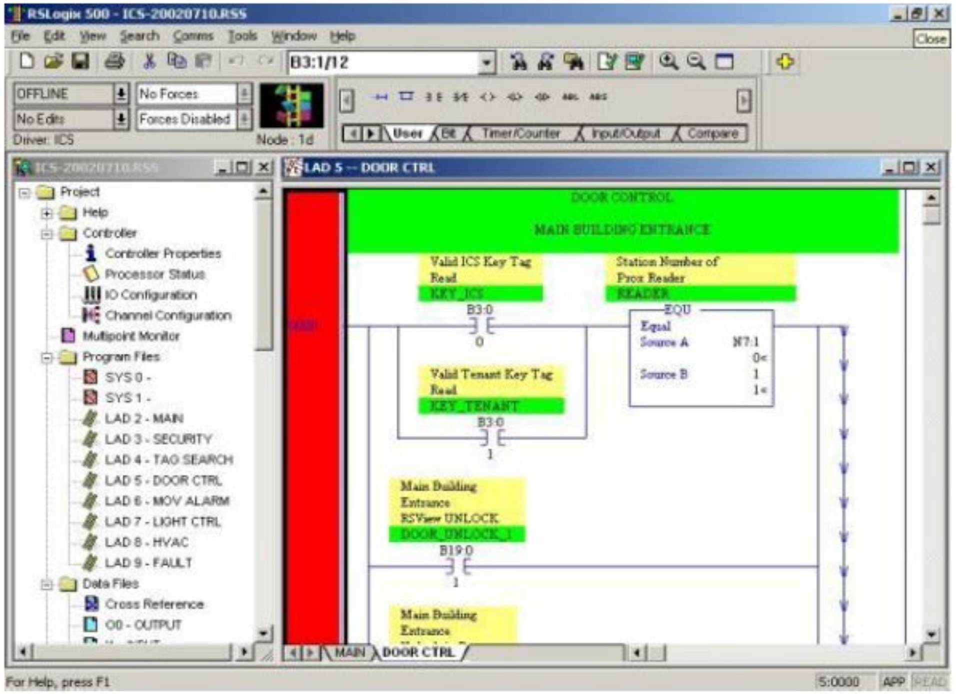
- Text-based languages:

  1. Instruction list - low-level computer language

  2. Structured text – high-level computer language



PLC Programming

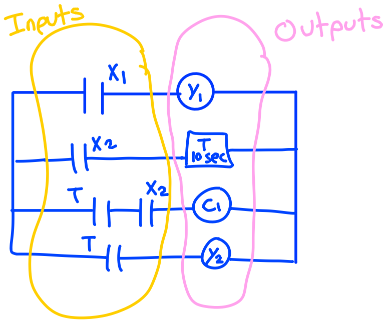


inputs
outputs
كيف يرتبطو مع بعض

أهم اشي تحديد

PLC Questions

An Industrial Robot



An industrial robot performs a machine loading and unloading operation. A PLC is used as the cell controller. The cell operates as follows:

(1) a human worker places a part into a nest,

(2) the robot reaches over and picks up the part and places it into an induction heating coil,

(3) a time of 10 sec is allowed for the heating operation, and (

4) the robot reaches into the coil, retrieves the part, and places it on an outgoing conveyor.

A limit switch X1 (normally open) is used to indicate that the part is in the nest in step (1). This energizes output contact Y1 to signal the robot to execute step (2) of the work cycle (this is an output contact for the PLC, but an input interlock signal for the robot controller).

A photocell X2 is used to indicate that the part has been placed into the induction heating coil C1.

Timer T1 is used to provide the 10-sec heating cycle in step (3), at the end of which, output contact Y2 is used to signal the robot to execute step (4).

Construct the ladder logic diagram for the system

* Inputs

X_1 : limit switch
part on the nest

X_2 : photo cell
part on heating coil

*Timer

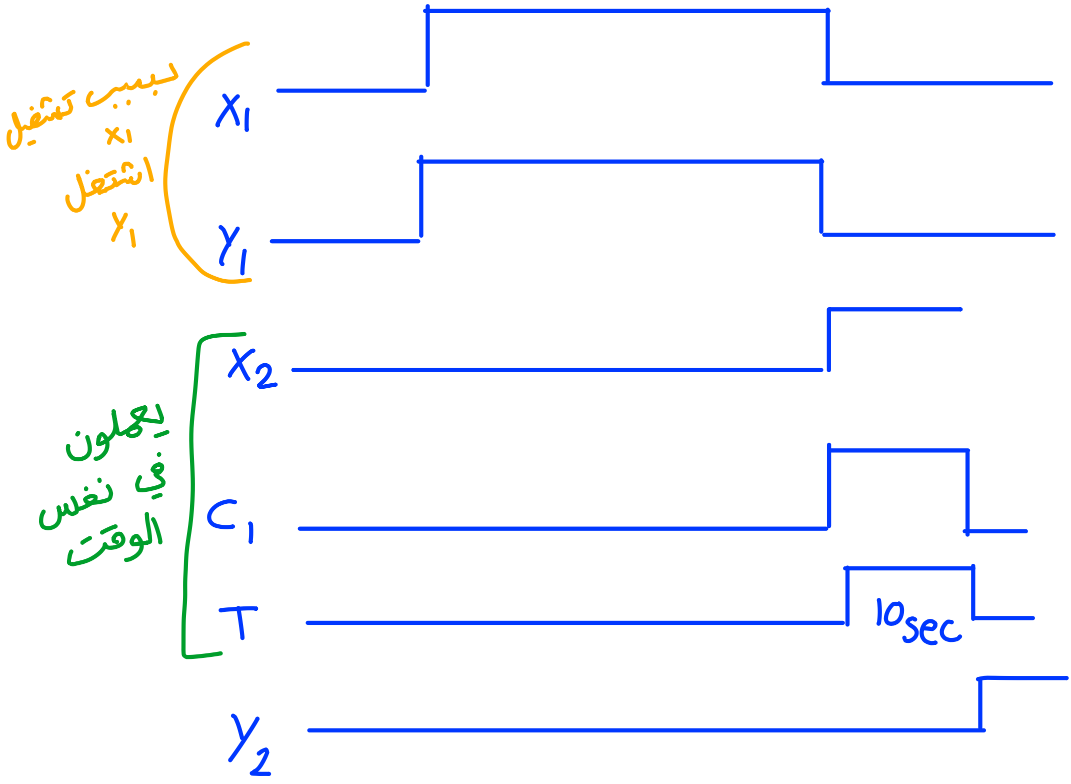
T : 10 sec

*outputs

Y_1 : Robot signal
nest → heating coil

Y_2 : Robot signal
heating coil → conuyer

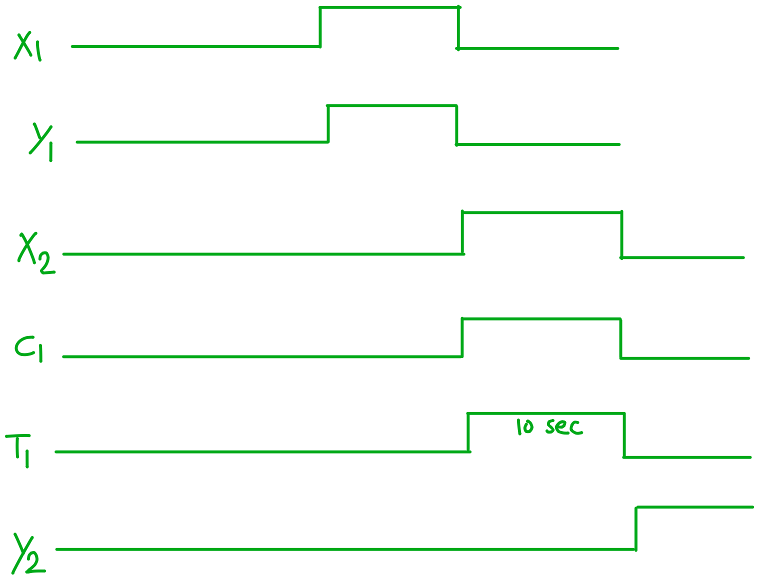
C_1 : heating coil



inputs → X_1, X_2

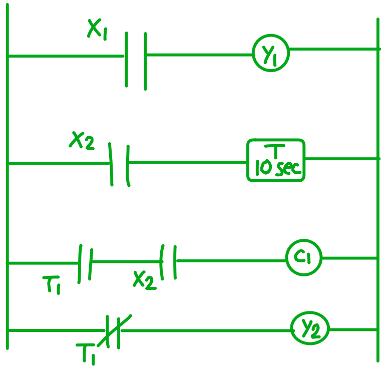
Timers → T_1

Outputs → Y_1 robot signal nest → heating coil
→ C_1 heating coil
→ Y_2 robot signal



⟹

Ladder diagram



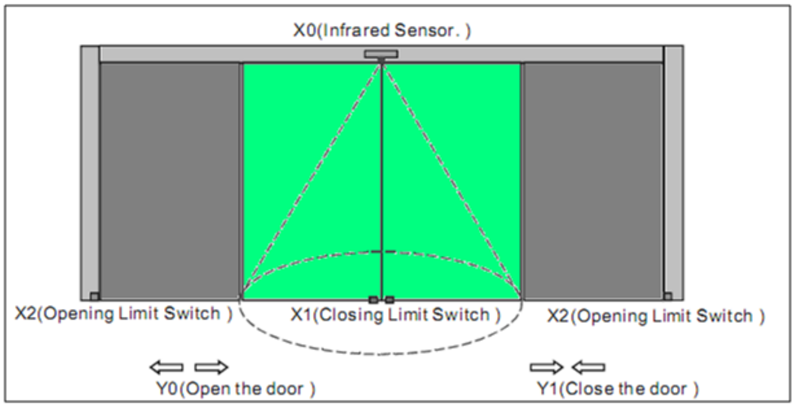
Inputs : X0
X2 open limit switch
X1 close limit switch

Time : T : 7 seconds

outputs : Y0 open the door
Y1 close the door

يفتح الباب motor
بغلق الباب motor
sensor
ينتظر و هو فاتح 7 seconds الا اذا حداجى time

Automatic opening closing Door PLC Program


X0(Infrared Sensor.)
X2(Opening Limit Switch)
X1(Closing Limit Switch)
X2(Opening Limit Switch)
Y0(Open the door)
Y1(Close the door)

When someone enters the infrared sensing field, opening motor starts working to open the door automatically till the door touches the opening limit switch If the door touches the opening limit switch for 7 sec and nobody

enters the sensing field, the closing motor starts working to close the door automatically till the closing limit switch touched together. Stop the closing action immediately if someone enters the sensing field during the door closing process.

Number of PLC Inputs Required

X0 – X0 = ON when someone enters the sensing field.

X1 – Closing limit switch. X1 = ON when 2 switches touched together.

X2 – Opening limit switch. X2 = ON when the door touched the switches.

Number of PLC Outputs Required

Y0 -Opening motor

Y1 – Closing motor

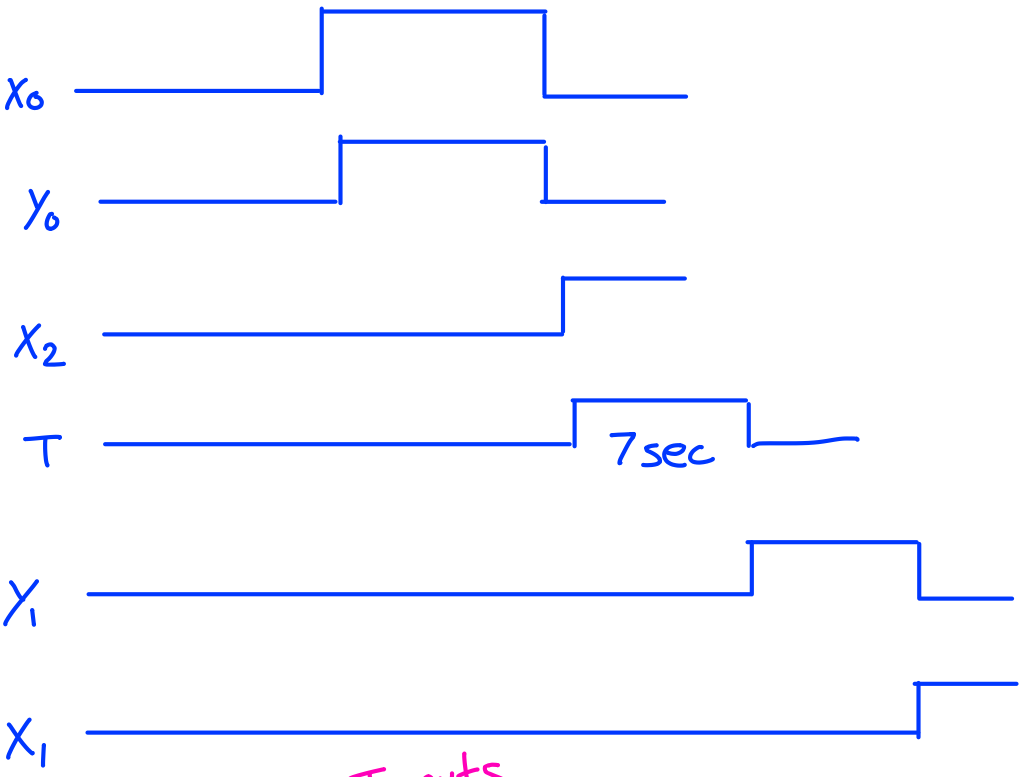
Number of PLC Timers Required

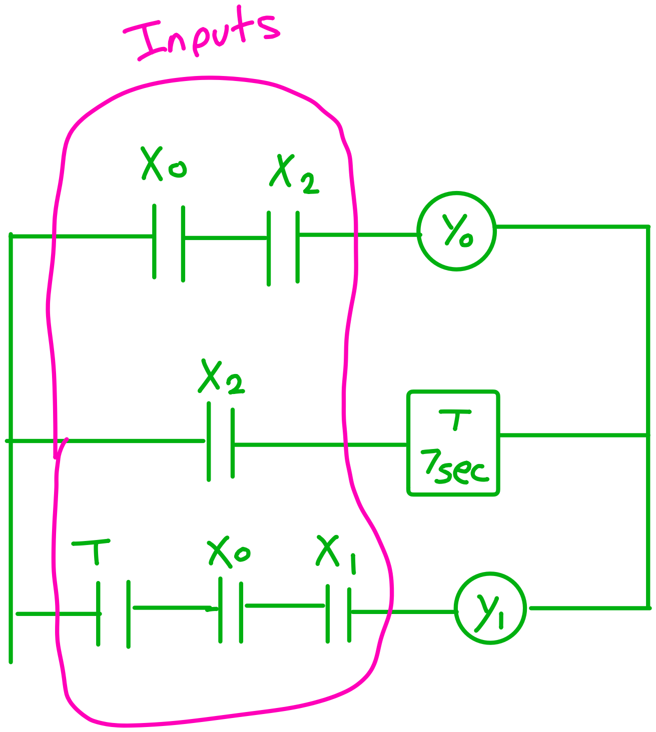
T0 – 7 sec timer.

Inputs : X_0
X_2 open limit switch
X_1 close limit switch

Time : T : 7 seconds

outputs : Y_0 open the door
Y_1 close the door





Consider the Automatic coffee maker as illustrated in Figure 1:

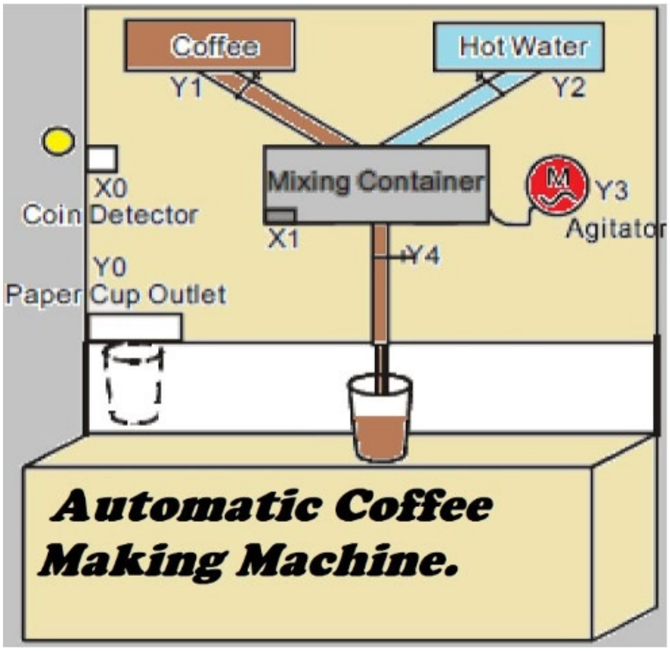
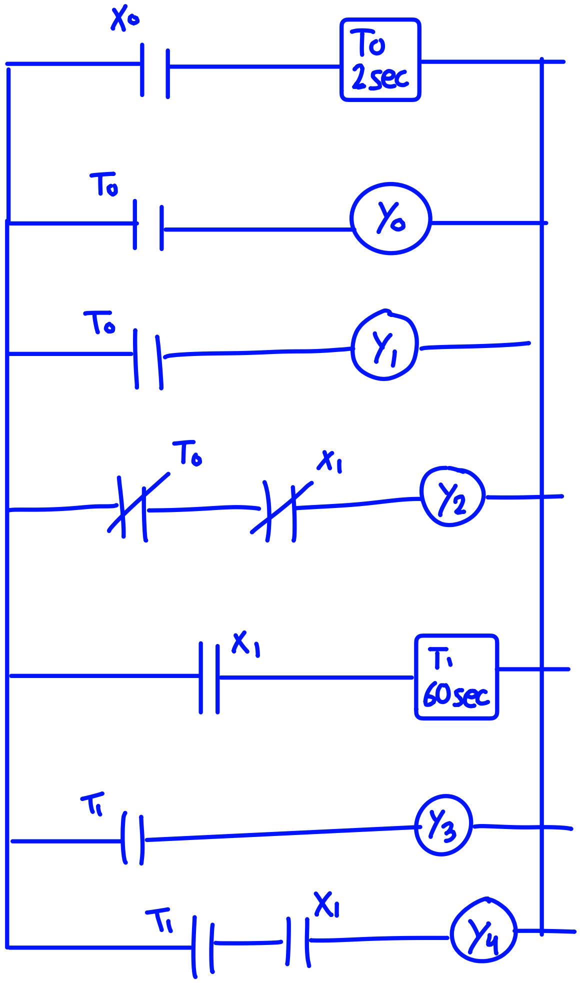


Fig.1 Automatic coffee maker

1. When a coin is inserted, ~~X1~~ اظن X0 is HIGH (ON) and the following outputs will be activated at the same time:
   - A timer T0 will be activated for 2 sec
   - Y0 (paper cup outlet) will be HIGH (ON) and latched (a paper cup will be sent out)
   - Y1 (coffee powder outlet) will be HIGH (ON) and latched (a certain amount of coffee will be poured into the container).
   - Y0 and Y1 will be HIGH (ON) for 2 sec, which is the set value of the timer T0.

2. After 2 sec, Y2 (hot water outlet) will be activated HIGH (ON), and the hot water will be poured in the container. At the same time, Y0 and Y1 will be closed LOW (OFF).
3. When the liquid in the container reaches a certain amount of pressure:
   - A pressure sensor X1 will be activated HIGH (ON).
   - Y2 will be reset LOW (OFF)
   - Timer T1 will be activated HIGH (ON) for 60 sec.
   - The agitator Y3 will be HIGH (ON) for 60 sec, which is the set value of Timer T1.
4. After 60 sec, the agitator Y3 will be Low (OFF) and Y4 (the ready – made coffee outlet) will be HIGH (ON) and latched and the ready –made coffee will be pouring out from the Y4 outlet.
5. When the coffee is poured into the paper cup completely, X1 will be LOW (OFF) and Y4 will be reset LOW (OFF) the ready-made coffee outlet will be closed.

Draw the PLC ladder diagram for the infusing container system above.


X0
T0
2sec
T0
Y0
T0
Y1
T0
X1
Y2
X1
T1
60sec
T1
Y3
T1
X1
Y4

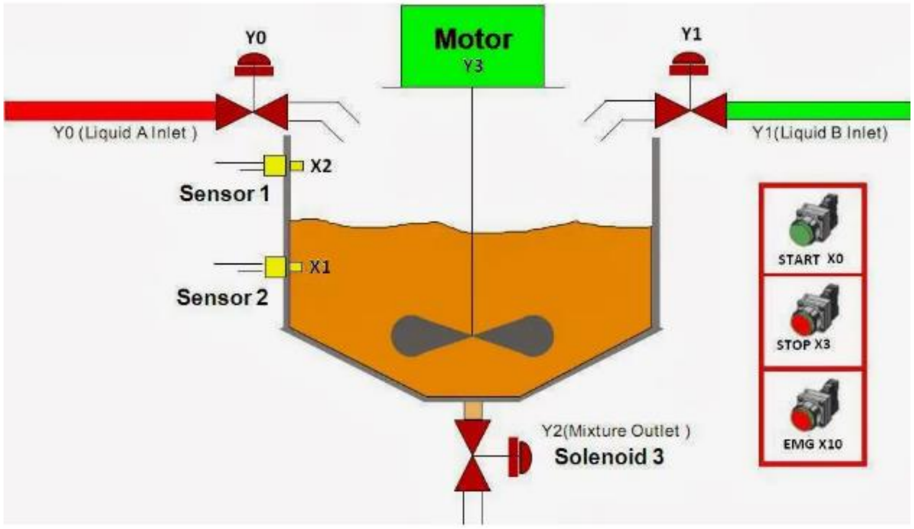
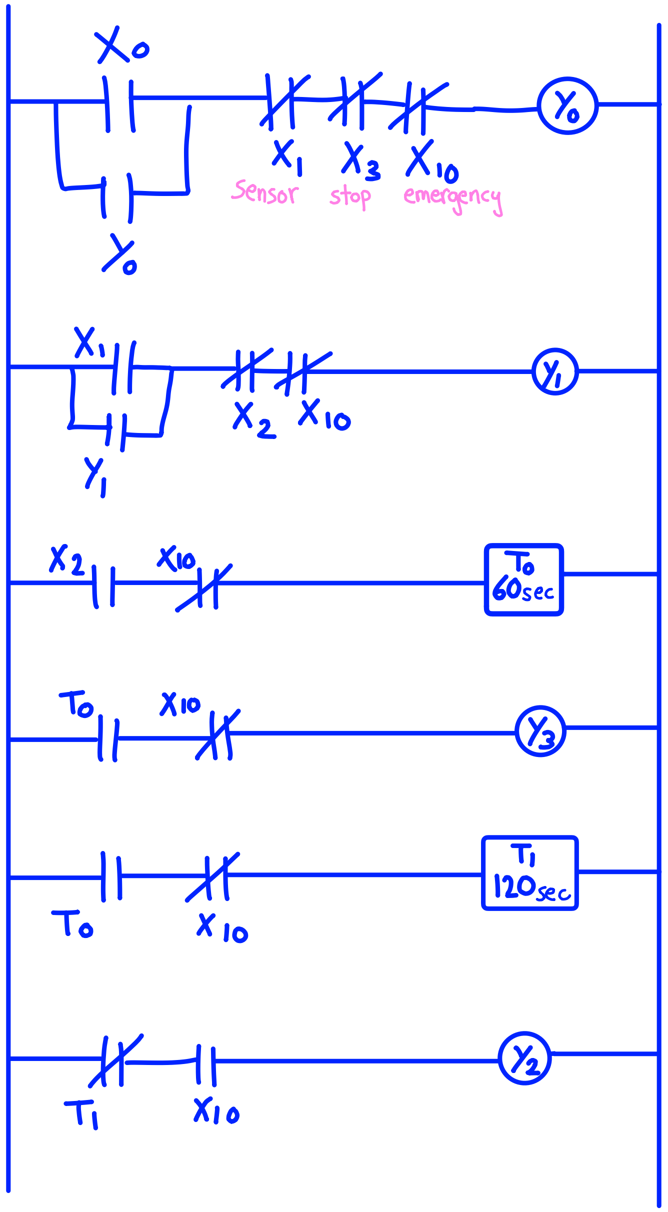


Fig.2

Consider the automatically infusing container with liquids A and B as illustrated in Figure 2:

1. When X0 (start button) will be ON when START is pressed. Y0 will be ON and latched, and the valve will be opened for infusing liquid A until the level reaches the low-level set point indicated by float sensor X1.
2. X1 will be ON when the level reaches the low-level float sensor. Y1 will be ON and latched, and the valve will be opened for infusing liquid B until the level reaches the high-level float sensor X2.
3. X2 will be ON when the level reaches the high-level float sensor. Y3 will be ON and activates the motor of the mixer. Also, timer T0 will be activated and start to count for 60 sec (mixing period).
4. After 60 sec, T0 will be OFF, and the mixer motor Y3 will stop working. Y2 will be ON , and the mixture will drain out of the container.
5. When Y2 = ON, timer T1 will be activated and start to count for 120 sec. After 120 sec, T1 will be Off and Y2 will be OFF. The draining process will be stopped.
6. When an error occurs, press EMERGENCY STOP button X10. The NC contact X10 will be ON to disable all the outputs. The system will then stop running.

Draw the PLC ladder diagram for the infusing container system above.



X_0
X_1
X_3
X_{10}
Y_0
Sensor
stop
emergency
Y_0
X_1
X_2
X_{10}
Y_1
Y_1
X_2
X_{10}
T_0
60sec
T_0
X_{10}
Y_3
T_0
X_{10}
T_1
120sec
T_1
X_{10}
Y_2