➢Chapter 3

➢Fundamentals of Logic

❑The Binary Concept

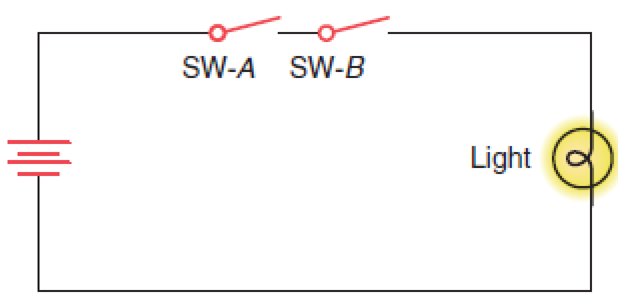
- The PLC, like all digital equipment, operates on the binary principle.
- The term *binary principle refers to the idea* that many things can be thought of as existing in only one of two states.
- These states are 1 and 0.
- The 1 and 0 can represent ON or OFF, open or closed, true or false, high or low, or any other two conditions.

❑AND, OR, and NOT Functions

➢ The basic rules that apply to an AND gate are:

- If all inputs are 1, the output will be 1.
- If any input is 0, the output will be 0.

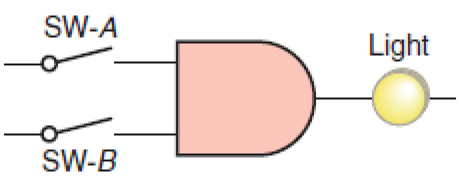
Hardwired circuit



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

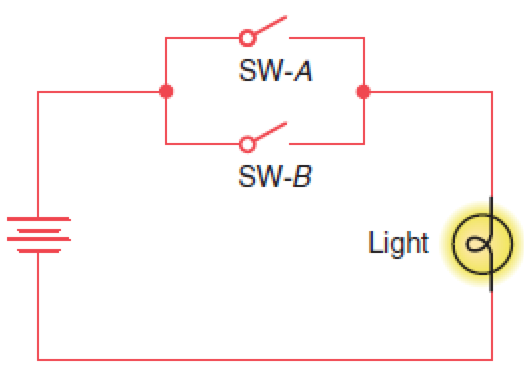


❑AND, OR, and NOT Functions

➢ The basic rules that apply to an OR gate are:

- If one or more inputs are 1, the output is 1.
- If all inputs are 0, the output will be 0.

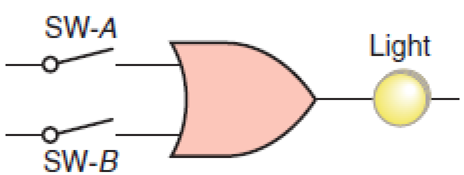
Hardwired circuit



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)

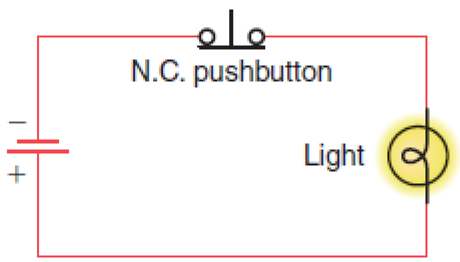
Logic representation



❑AND, OR, and NOT Functions

➢ The logical NOT function can be performed on a contact input simply by using a normally closed instead of a normally open contact

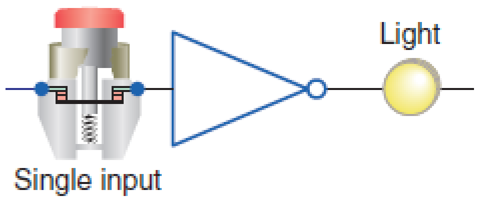
Hardwired circuit



Truth table

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

Logic representation

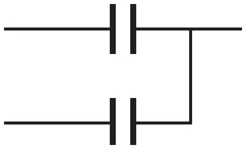


❑Boolean Algebra

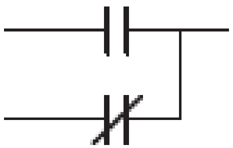
normally open contact

normally closed contact

Or (OR)



Or Not (OR NOT

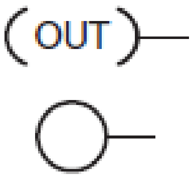


And (AND)

❑Boolean Algebra

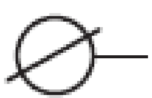
And not

Out (OUT)



Or Out (OR OUT)

Output Not (OUT NOT)



❑Hardwired Logic versus Programmed Logic

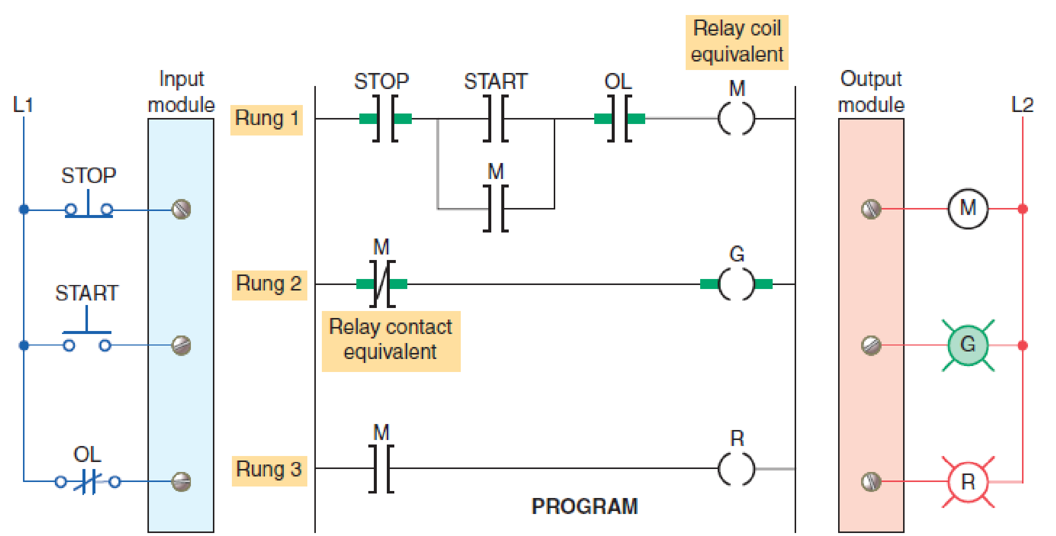
➢The term *hardwired logic refers to logic control functions* that are determined by the way devices are electrically interconnected.

➢Hardwired logic can be implemented using relays and relay ladder schematics.

➢Relay ladder schematics are universally used and understood in industry.

➢Figure shows a typical relay ladder schematic of a motor stop/start control station with pilot lights

➢The instructions used are the relay equivalent of normally open (NO) and normally closed (NC) contacts and coils..



❑Hardwired Logic versus Programmed Logic

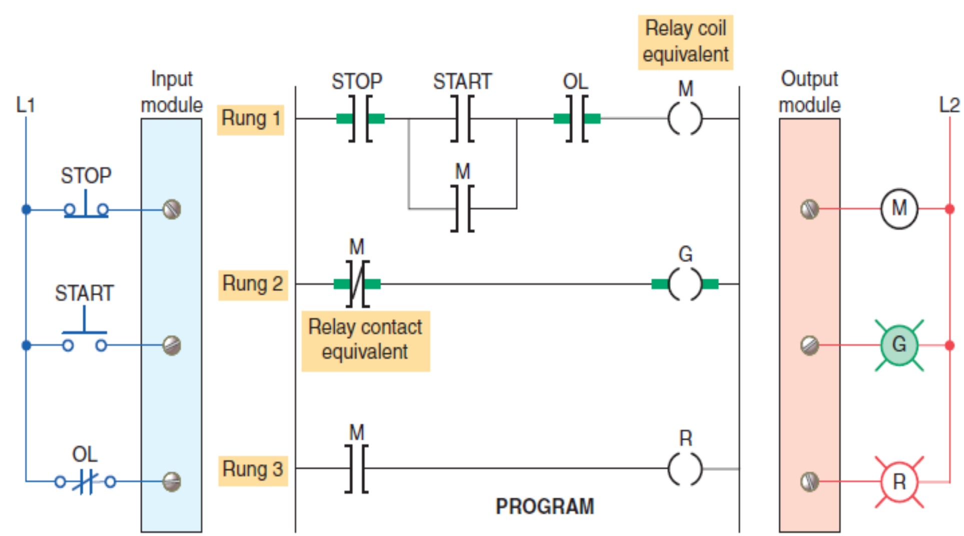
➢PLC contact symbolism is a simple way of expressing the control logic in terms of symbols.

➢ These symbols are basically the same as those used for representing hardwired relay control circuits.

➢A rung is the contact symbolism required to control an output.

➢Some PLCs allow a rung to have multiple outputs while others allow only one output per rung.

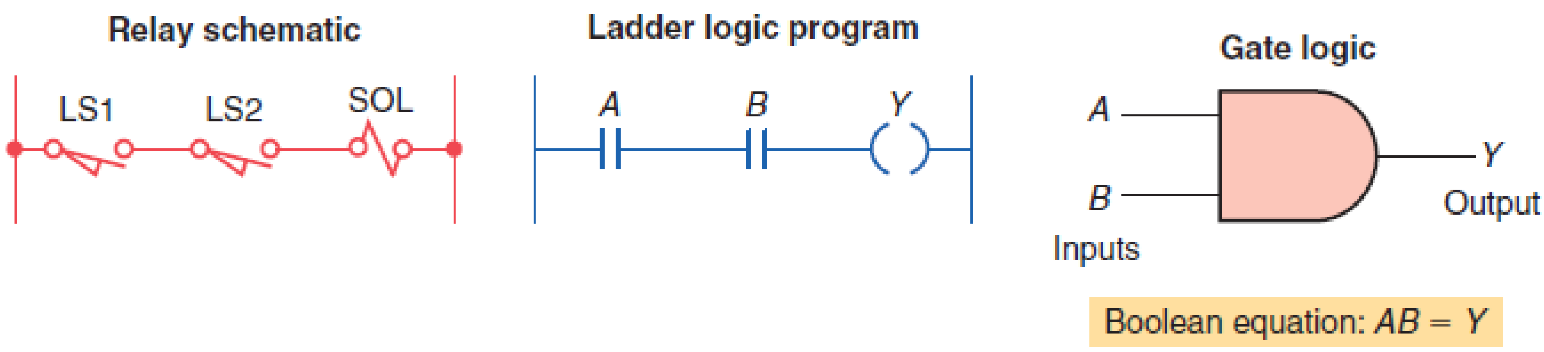
➢A complete ladder logic Program then consists of several rungs, each of which controls an output.

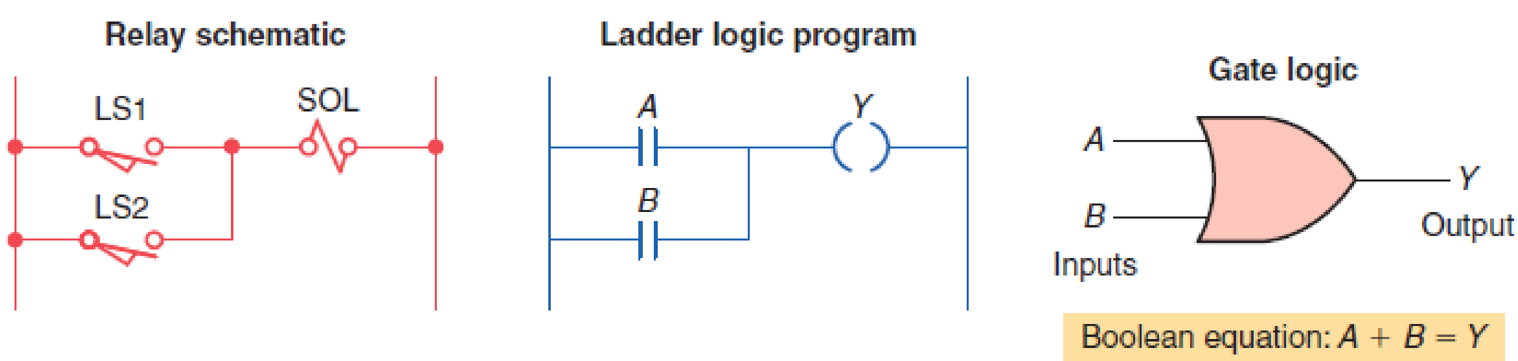


❑Hardwired Logic versus Programmed Logic

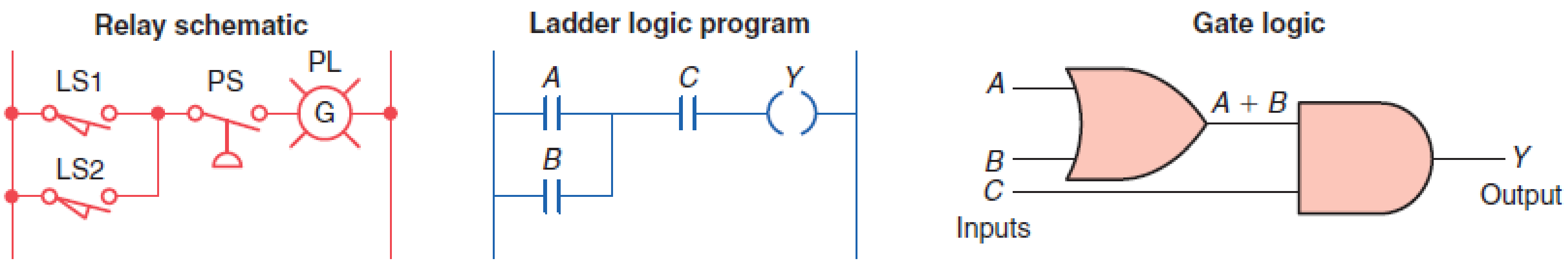
- In programmed logic all mechanical switch contacts are represented by a software contact symbol and all electromagnetic coils are represented by a software coil symbol.
- Because the PLC uses ladder logic diagrams, the conversion from any existing relay logic to programmed logic is simplified.
- Each rung is a combination of input conditions (symbols) connected from left to right, with the symbol that represents the output at the far right.
- The symbols that represent the inputs are connected in series,parallel, or some combination of the two to obtain the desired logic.
- The following group of examples illustrates the relationship between the relay ladder schematic, the ladder logic program, and the equivalent logic gate circuit.

❑Hardwired Logic versus Programmed Logic

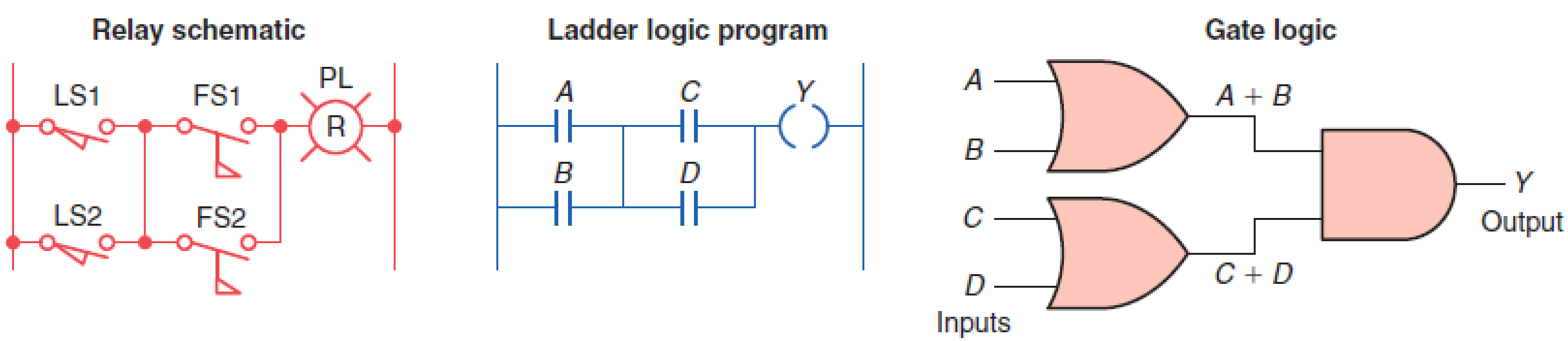




❑Hardwired Logic versus Programmed Logic

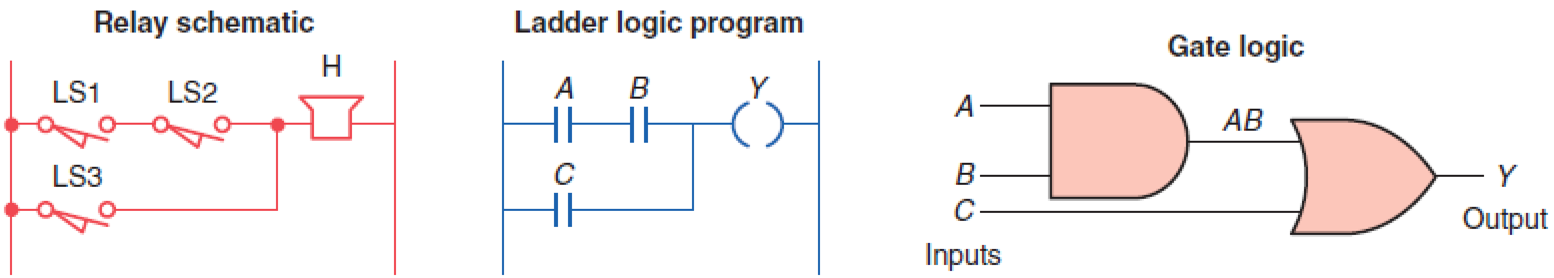


Boolean equation: $(A + B)C = Y$

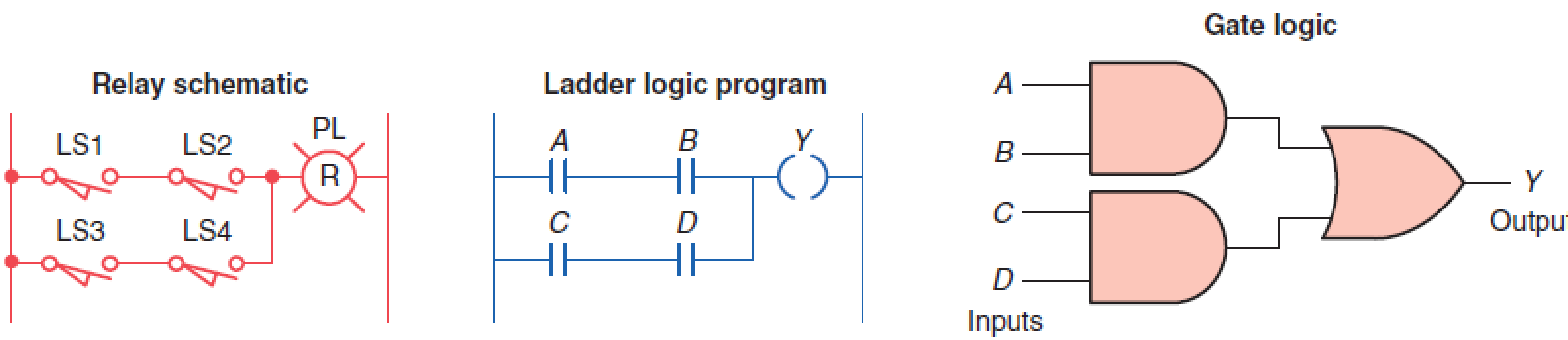


Boolean equation: $(A + B)\,(C + D) = Y$

❑Hardwired Logic versus Programmed Logic

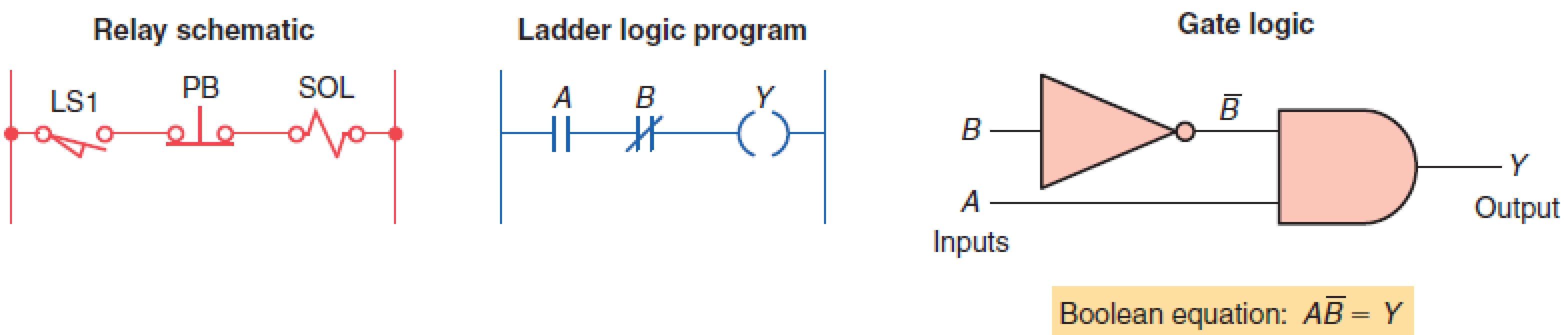


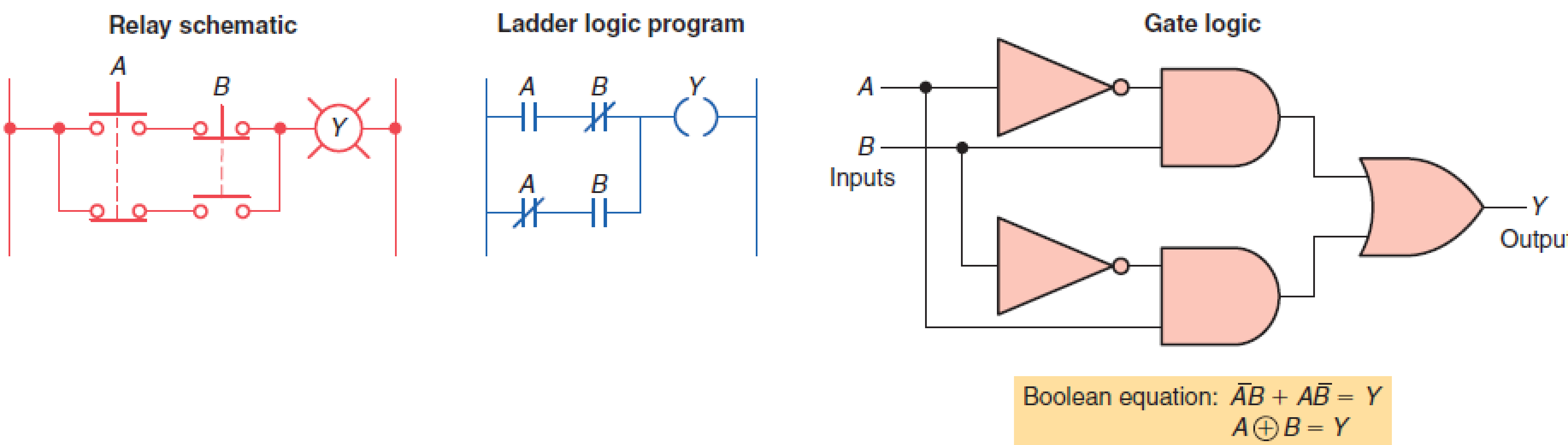
Boolean equation: $(AB) + C = Y$



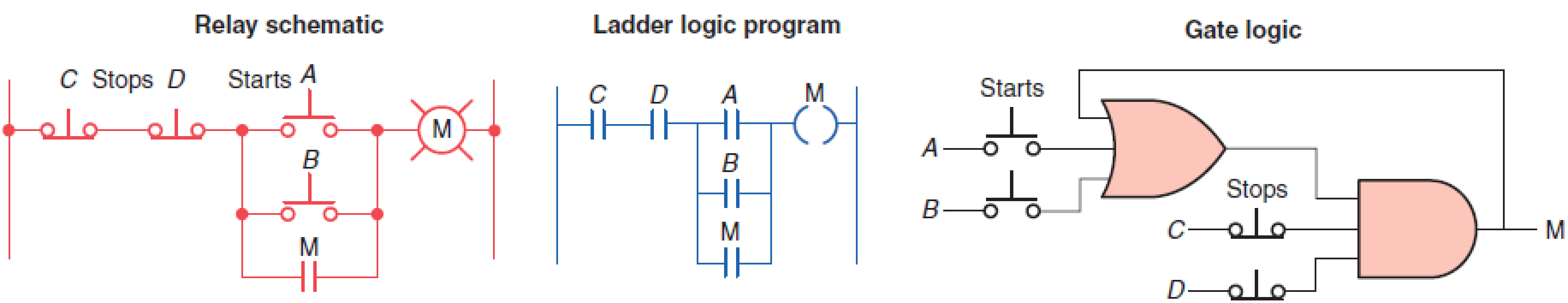
Boolean equation: $(AB) + (CD) = Y$

❑Hardwired Logic versus Programmed Logic



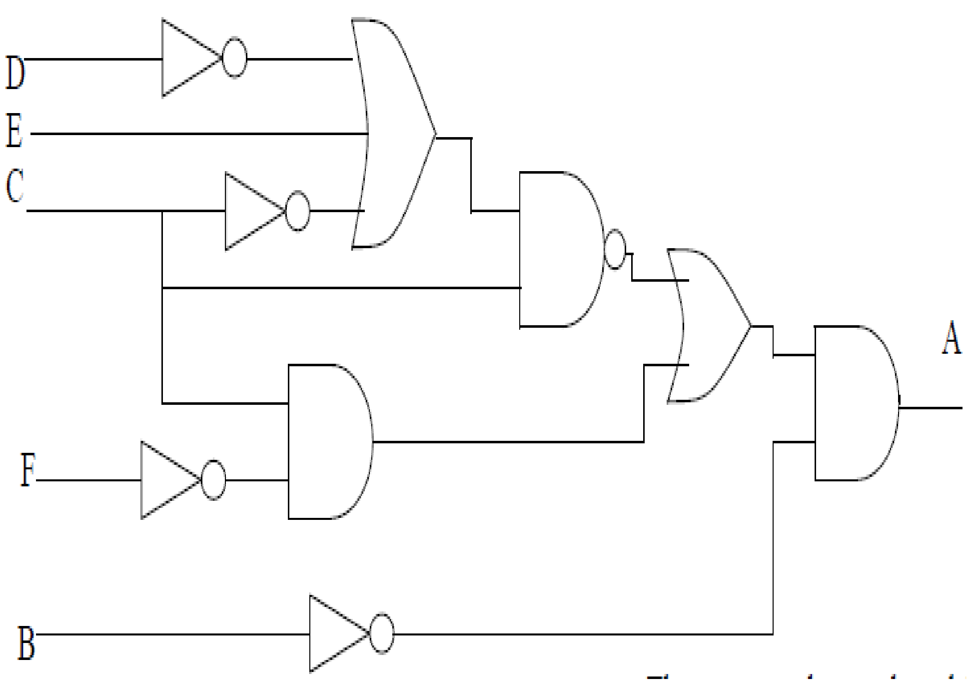


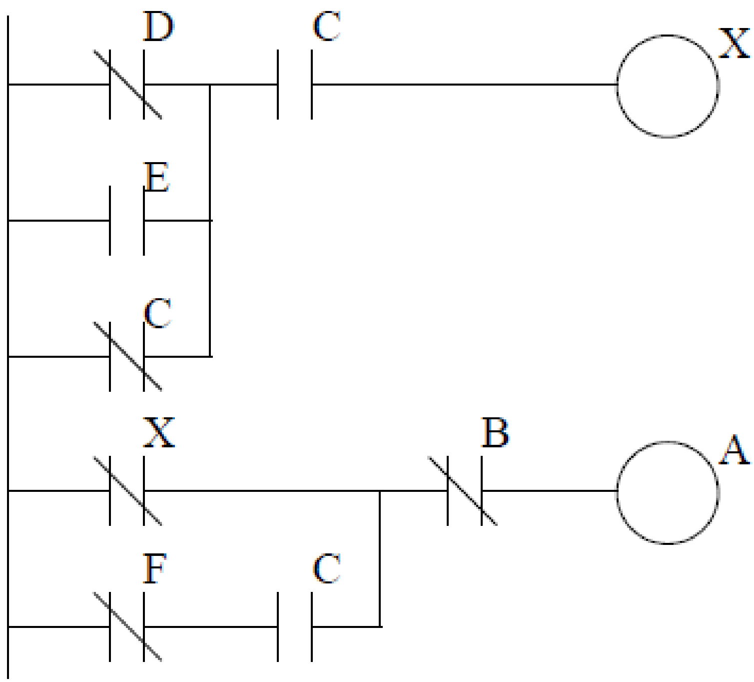
❑Hardwired Logic versus Programmed Logic



❑Hardwired Logic versus Programmed Logic

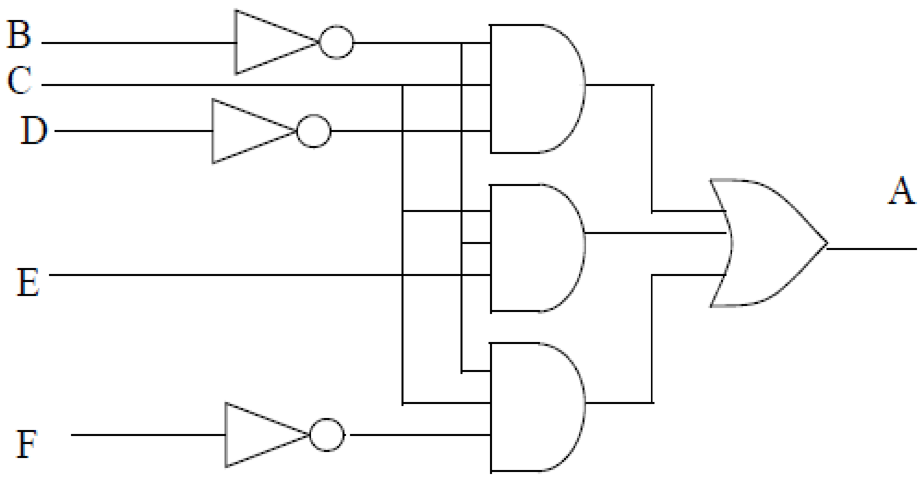
e.g :- $A = \bar{B} \cdot (\overline{C \cdot (\bar{D} + E + \bar{C})} + \bar{F} \cdot C)$

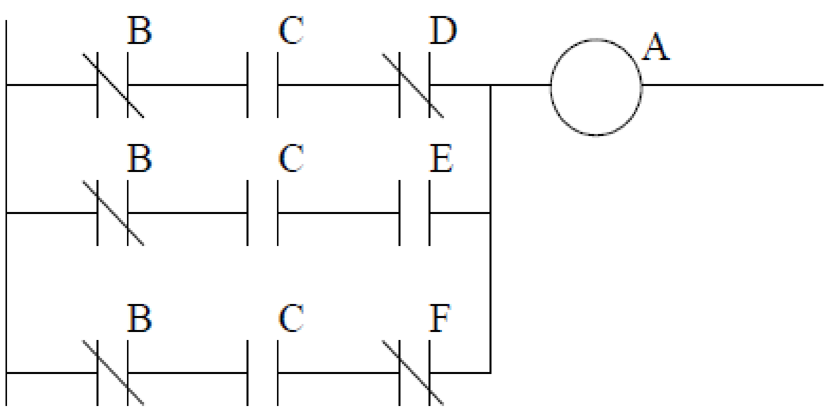




❑Hardwired Logic versus Programmed Logic

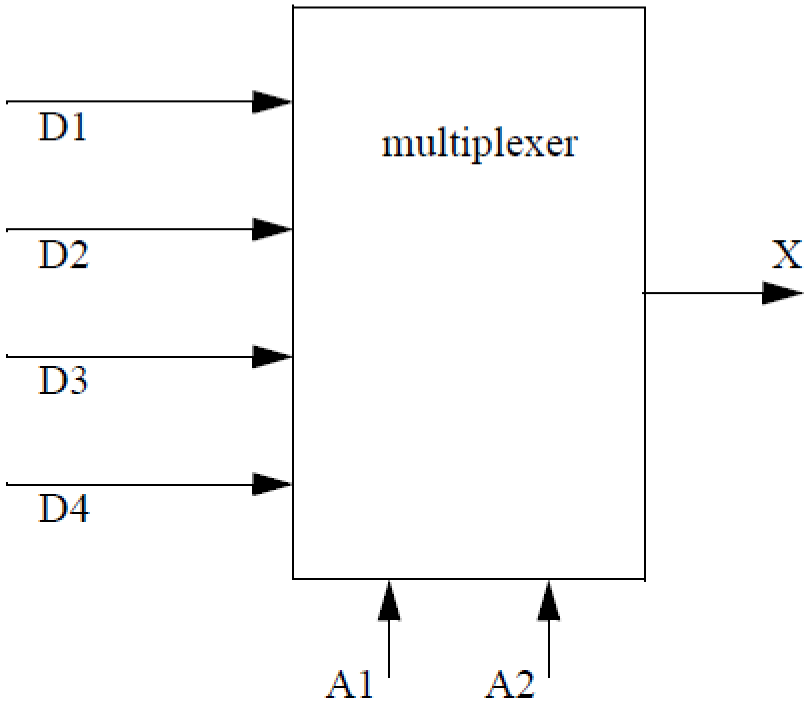
e.g :- $A = (\overline{B} \cdot C \cdot \overline{D}) + (\overline{B} \cdot C \cdot E) + (\overline{B} \cdot C \cdot \overline{F})$





❑Hardwired Logic versus Programmed Logic

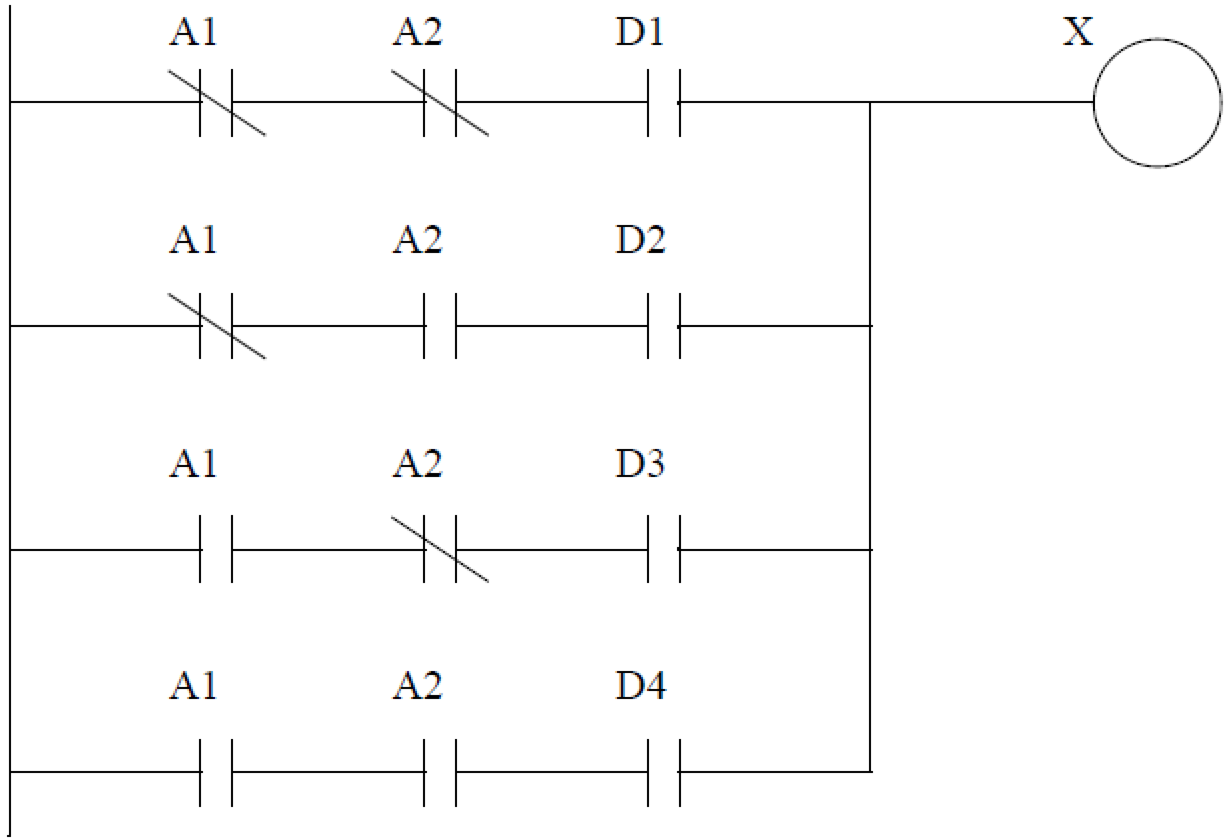
e.g :-



A1	A2	X
0	0	X=D1
0	1	X=D2
1	0	X=D3
1	1	X=D4

❑Hardwired Logic versus Programmed Logic

e.g :-



❑Hardwired Logic versus Programmed Logic

e.g :- A Burglar Alarm

Consider the design of a burglar alarm for a house. When activated an alarm and lights will be activated to encourage the unwanted guest to leave. This alarm be activated if an unauthorized intruder is detected by window sensor and a motion detector. The window sensor is effectively a loop of wire that is a piece of thin metal foil that encircles the window. If the window is broken, the foil breaks breaking the conductor. This behaves like a normally closed switch. The motion sensor is designed so that when a person is detected the output will go on. As with any alarm an activate/deactivate switch is also needed. The basic operation of the alarm system, and the inputs and outputs of the controller are itemized

❑Hardwired Logic versus Programmed Logic

e.g :- A Burglar Alarm

The inputs and outputs are chosen to be;
A = Alarm and lights switch (1 = on)
W = Window/Door sensor (1 = OK)
M = Motion Sensor (0 = OK)
S = Alarm Active switch (1 = on)
The basic operation of the alarm can be described with rules.
1. If alarm is on, check sensors.
2. If window/door sensor is broken (turns off), sound alarm and turn on Lights

❑Hardwired Logic versus Programmed Logic

e.g :- A Burglar Alarm

Inputs			Output
S	M	W	A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

alarm off

alarm on/no thief

alarm on/thief detected

note the binary sequence

❑Hardwired Logic versus Programmed Logic

e.g :- A Burglar Alarm

$$A = (S \cdot \overline{M} \cdot \overline{W}) + (S \cdot M \cdot \overline{W}) + (S \cdot M \cdot W)$$

$$\therefore A = S \cdot (\overline{M} \cdot \overline{W} + M \cdot \overline{W} + M \cdot W)$$

$$\therefore A = S \cdot ((\overline{M} \cdot \overline{W} + M \cdot \overline{W}) + (M \cdot \overline{W} + M \cdot W))$$

$$\therefore A = (S \cdot \overline{W}) + (S \cdot M) = S \cdot (\overline{W} + M)$$