# 1. Introductions

## Part C

Logic Gates: building blocks for digital circuits, perform basic logical functions based on Boolean algebra in binary condition

AND: output is “true” when both inputs are “true” otherwise “false” aka Conjunction

OR: output is “true id either or both inputs are “true” otherwise “false” aka Disjunction

NOT: logical inverter, reverses the input and output aka Negation/ Complement

NAND: AND gate followed by NOT gate; output is “false” if both inputs are “true” else “true”

NOR: OR gate followed by NOT gate; output is “true” if both inputs are “false” else “false

XOR: exclusive-OR act as either/or; output is “true” if either, but not both, the input is “true” and output is “false” if both inputs are “false” or if both inputs are “true”

XNOR: exclusive-NOR, combination of XOR gate followed by NOT gate. Output is “true if the inputs are the same, and “false” if inputs are different.

Boolean Algebra: branch category of algebra where it deals with truth values, true (1) and false (0) to simplify digital circuits and gates. or Binary or Logical Algebra. Fundamental component of digital electronics

Boolean Variable: Alphabetical letters used to define a symbol to represent logical quantities

Truth Table: table that gives all possible values of logical combinations. Number of rows should be equal to $2^n$, where “n” is the number of variables/inputs.

Boolean Algebra Theorems: de Morgan’s First and Second Law which reduce expression in simplified form by changing its form

1st: Complement of Product is equal to Sum of Individual Complements

2nd: Complement of Sum is equal to Product of Individual Complements

Boolean Theorem/Laws:

| | | |
|---|---|---|
| 1a. $X \cdot 0 = 0$ | 1b. $X + 1 = 1$ | Annulment Law |
| 2a. $X \cdot 1 = X$ | 2b. $X + 0 = 0$ | Identity Law |
| 3a. $X \cdot X = X$ | 3b. $X + X = X$ | Idempotent Law |
| 4a. $\overline{X} \cdot X = 0$ | 4b. $\overline{X} + X = 0$ | Complement Law |
| 5a. $\overline{\overline{X}} = X$ | | Double Negation Law |
| 6a. $X \cdot Y = Y \cdot X$ | 6b. $X + Y = Y + X$ | Commutative Law |
| 7a. $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z = (X \cdot Z) = Y = XYZ$ | | Associative Law |
| 7b. $X + (Y + Z) = (X + Y) + Z = (X + Z) + Y = X + Y + Z$ | | |
| 8a. $X \cdot (Y + Z) = XY + XZ$ | | Distributive Law |
| 8b. $X + YZ = (X + Y) \cdot (X + Z)$ | | |
| 9a. $\overline{X \cdot Y} = \overline{X} + \overline{Y}$ | 9b. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$ | de Morgan's Theorem |
| 10a. $X \cdot (X + Y) = X$ | 10b. $X + XY = X$ | Absorption Law |
| 11a.$(X + Y) \cdot (X + \bar{Y}) = X$ | 11b. $XY + X\bar{Y} = X$ | Redundancy Law |
| 12a. $(X + \overline{Y}) \cdot Y = XY$ | 12b. $X\overline{Y} + Y = X + Y$ | |
| 13a. $(X + Y) \cdot (\bar{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\bar{X} + Z)$ | | Consensus Law |
| 13b. $XY + \bar{X}Z + YZ = XY + \bar{X}Z$ | | |
| 14a. $X \oplus Y = (X + \overline{Y}) \cdot (\bar{X} + Y)$ | | XOR Gate |
| 14b. $X \oplus Y = XY + XY$ | | |
| 15a. $X \odot Y = (X + Y) \cdot (\bar{X} \cdot \bar{Y})$ | | XNOR Gate |
| 15b. $X \odot Y = \overline{X}\,\overline{Y} + X\,Y$ | | |
| 15c. $X \odot Y = (X + Y) \cdot (\overline{X} + \overline{S})$ | | |

## 2.3. Part C

### 2.3.1. Siren Gloves

A factory-producing gloves sends the signal for quitting time through the sound of the siren. The siren should be activated when either of the following conditions is met:

- It is after 10 pm and all machines are shut down,
- It is Monday, the production run for the day is complete and all machines are shut down.

Design a logic circuit that will control the siren. (Use four logic input variables to represent the various conditions, for example, input A will be HIGH only when the time of day is 10pm o'clock or later)

We have used four input variables:

A: represents the time of day. A is HIGH when it is 10 pm or later.

B: represents machines shut down. B is HIGH when all machines are shut down.

C: represents the day of the week. C is HIGH when it is Monday.

D: represents the production run. D is HIGH when the production run is complete.

Y: Siren sounds the alarm. High when siren is on.

Truth Table:

| **A** | **B** | **C** | **D** | **Y** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | **1** |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | **1** |
| 1 | 1 | 0 | 1 | **1** |
| 1 | 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | 1 | **1** |

Conditions in which siren will sound i.e., AND (product) term for each case where the output is 1 are given by:

- A’. B. C. D: All machines shut down, Monday, and the production completed.
- A. B. C’. D’: After 10 pm, all the machines shut down.
- A. B. C. D’: After 10 pm, all the machines shut down.
- A. B. C. D: After 10 pm, all the machines shut down.

The sum of product (SOP) when the siren is on is A’BCD + ABC’D’ + ABCD’ + ABCD

The simplified logical expression that will make the siren is on :

= A’BCD + ABC’D’ +ABCD’+ABCD

= AB (C’D’ +CD) +AB (C’D +CD’) + A’BCD

= AB [(C $\odot$ D) + (C $\oplus$ D)} + A’BCD

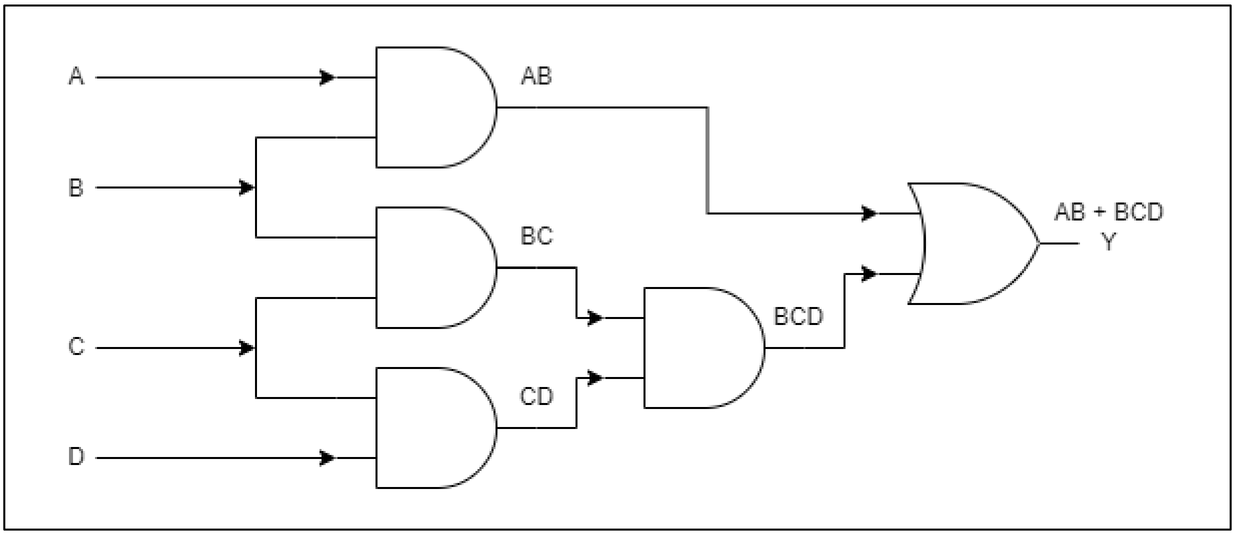
= AB+ A’BCD

= B(A+A’CD)

= B (A + CD)

= AB + BCD

Therefore, the simplified logical equation is AB + BCD .



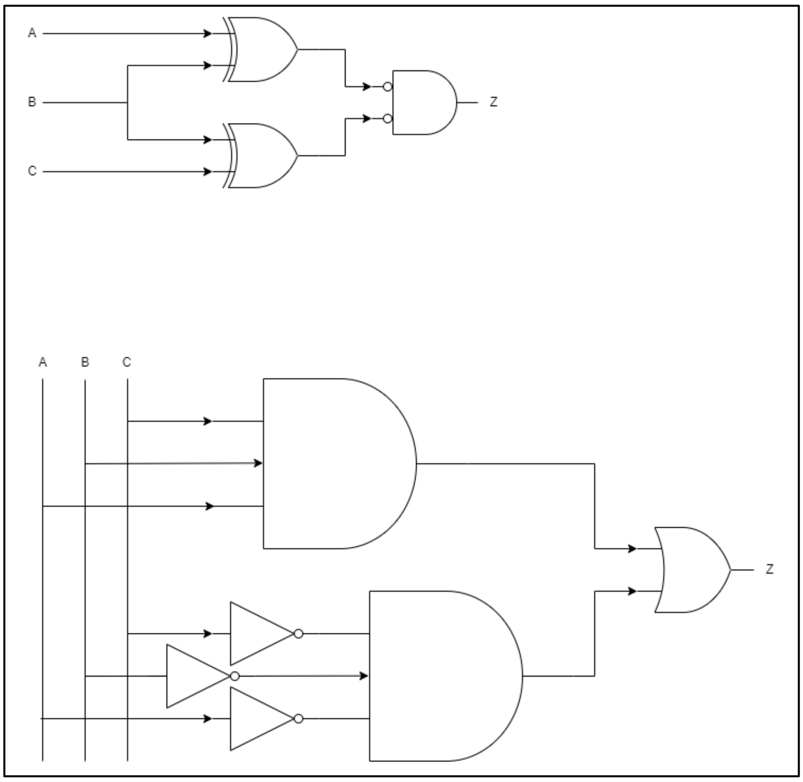
## 2.3.2. Same Level High Output

Design a circuit that produces a HIGH out only when all three inputs are at the same level.

A circuit that produces a HIGH out only when all three inputs are at the same level.

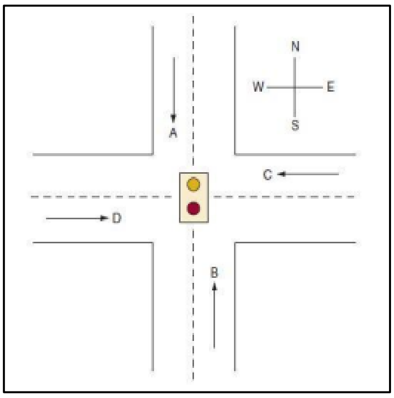
| INPUT | | | OUTPUT | |
|---|---|---|---|---|
| A | B | C | Z | AND |
| 0 | 0 | 0 | 1 | $\bar{A}\bar{B}\bar{C}$ |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 1 | ABC |

Sum of Products: Z= ABC + $\bar{A}\bar{B}\bar{C}$ (can't simplify anymore)



### 2.3.3. 4-Lane Traffic Control

The picture below shows the intersection of a road. Some vehicle detection sensors are placed along lanes C and D (main road) and lanes A and B (access road). These sensors' outputs are LOW (0) when no vehicle is present and HIGH (1) when a vehicle is present. The intersection traffic light is to be controlled according to the following logic:




A. The east-west (E-W) traffic light will be green whenever both lanes C and D are occupied.
B. The E-W light will be green whenever either C or D is occupied but lanes A and B are not both occupied.
C. The N-S light will also be green whenever both lanes A and B are occupied but C and D are not both occupied.
D. The N-S light will also be green when either A or B is occupied while C and D are both vacant.
E. The E-W light will be green when no vehicles are present.

Using the sensor outputs, A, B, C and D as inputs, design a logic circuit to control the traffic light. There should be two outputs-S and E-W that go HIGH when the corresponding light is to be green. Simplify the circuit as much as possible and show all steps.

OUTPUTS are HIGH (1) when there are vehicles on the road. LOW (0) when the road is empty

| INPUTS | | | | OUTPUTS (AND terms) | | | |
|---|---|---|---|---|---|---|---|
| A | B | C | D | E-W | | N-S | |
| 0 | 0 | 0 | 0 | 1 | $\bar{A}\bar{B}\bar{C}\bar{D}$ | 0 | |
| 0 | 0 | 0 | 1 | 1 | $\bar{A}\bar{B}\bar{C}D$ | 0 | |
| 0 | 0 | 1 | 0 | 1 | $\bar{A}\bar{B}C\bar{D}$ | 0 | |
| 0 | 0 | 1 | 1 | 1 | $\bar{A}\bar{B}CD$ | 0 | |
| 0 | 1 | 0 | 0 | 0 | → | 1 | $\bar{A}B\bar{C}\bar{D}$ |
| 0 | 1 | 0 | 1 | 1 | $\bar{A}B\bar{C}D$ | 0 | |
| 0 | 1 | 1 | 0 | 1 | $\bar{A}BC\bar{D}$ | 0 | |
| 0 | 1 | 1 | 1 | 1 | $\bar{A}BCD$ | 0 | |
| 1 | 0 | 0 | 0 | 0 | → | 1 | $A\bar{B}\bar{C}\bar{D}$ |
| 1 | 0 | 0 | 1 | 1 | $A\bar{B}\bar{C}D$ | 0 | |
| 1 | 0 | 1 | 0 | 1 | $A\bar{B}C\bar{D}$ | 0 | |
| 1 | 0 | 1 | 1 | 1 | $\bar{A}B\bar{C}\bar{D}$ | 0 | |
| 1 | 1 | 0 | 0 | 0 | → | 1 | $AB\bar{C}\bar{D}$ |
| 1 | 1 | 0 | 1 | 0 | | 1 | $AB\bar{C}D$ |
| 1 | 1 | 1 | 0 | 0 | | 1 | $ABC\bar{D}$ |
| 1 | 1 | 1 | 1 | 1 | $ABCD$ | 0 | |

Looking at the Truth Table above. $N - S$ is opposite of $E - S$; $E - S = \overline{(N - S)}$

Since there are only five cases when $N - S = 1$

Sum of Products: $N - S$

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

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

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

$$= \overline{CD}(A \oplus B) + AB(C \oplus D) + AB\bar{C}\bar{D}$$

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

Sum of Product: $E - S$

$= \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + ABCD$

$=(\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D}) + (\overline{A}B\overline{C}D + \overline{A}BC\overline{D}) + (A\overline{B}\overline{C}D + A\overline{B}C\overline{D}) + (\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D}) + (\overline{A}\overline{B}CD + \overline{A}BCD)$
$+ABCD$

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

$= \overline{A}\overline{B}(C \oplus D) + \overline{A}B(C \oplus D) + A\overline{B}(C \oplus D) + (\overline{A}\overline{B} + \overline{A}B)(\overline{C}\overline{D} + CD) + ABCD$

$= (C \oplus D)(\overline{A}B + A\overline{B}) + \overline{A}\overline{B}(C \oplus D) + (\overline{A}\overline{B} + \overline{A}B)(1) + ABCD$

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

$= (C \oplus D)(A \oplus B + \overline{A}\overline{B}) + \overline{A}(1) + ABCD$

$\therefore \overline{A} + (C \oplus D)(A \oplus B + \overline{A}\overline{B}) + ABCD$