



# Control System Training

## MODULE 4 – Combinatorial Boolean Logic

# Copyright Notice

These training materials, including the samples, exercises, and solutions, are copyrighted materials. Any reproduction, or use of any kind without the specific written approval of the author is strictly prohibited.

Permission for extra-curricular use by First FRC teams for FRC related training is granted, provided the original copyright and acknowledgements are retained.

© Jim Simpson, 2018

# Combinatorial Boolean Logic

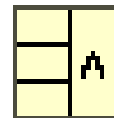
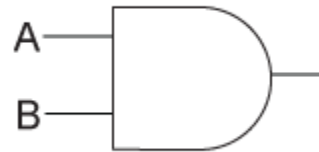
## Definitions:

- **Boolean** – Only values are: ZERO / ONE or TRUE/FALSE
- **Combinatorial Logic** – Outcome depends only on the current value of the inputs. Nothing is depends on time (or previous values of the inputs or outputs).

# AND Gate

AND GATE

Input		Output
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

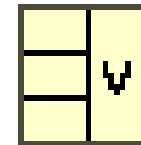


- Output is true when all inputs are true.
- Boolean algebra written as  $AB$  or  $A \text{ AND } B$

# OR Gate

OR GATE

Input		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

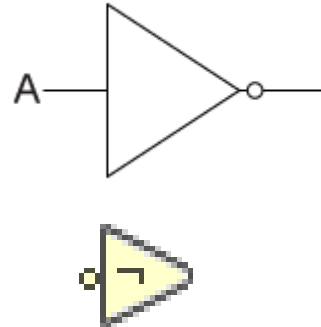


- Output is true when any input is true
- Boolean algebra written as  $A+B$

# NOT Gate

NOT GATE

INPUT	Output
0	1
1	0

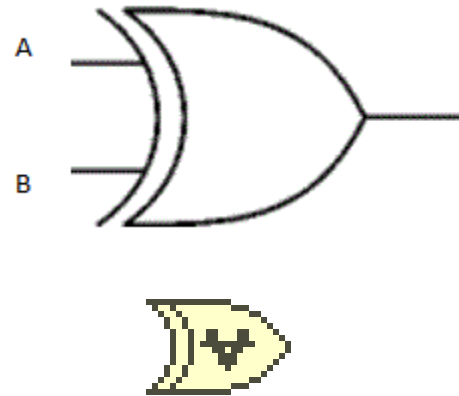


- Output is the opposite of the input.
- Boolean algebra written as  $\neg A$  or  $\overline{A}$

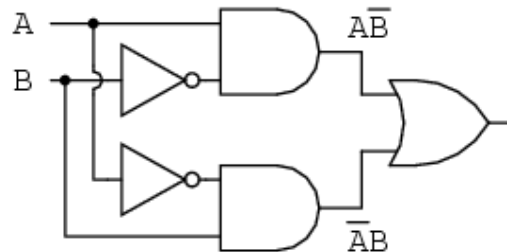
# XOR (Exclusive OR) Gate

XOR GATE

Input		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0



- Output is true when inputs are different.
- Boolean algebra written as  $A \oplus B$
- Equivalent to:



# Other Gates

- **NAND – Not And**
- **NOR – Not Or**
- **Not XOR – Not Exclusive OR**






# Boolean Algebra

- **Write out as regular algebra. Similar to numeric expression**
- **Do this as part of the simplification process**
- **Symbols**
  - + means OR
  - \* means AND
  - • means AND
  - Line over term or ^ means NOT
  - $\oplus$  means XOR (exclusive OR)
  - = means equals
- **Examples:**
  - $OUT = PERM \cdot INPUT$
  - $E = A \cdot B + C \cdot D$

# Boolean Algebra - Rules

1a. $X \cdot 0 = 0$	1b. $X + 1 = 1$
2a. $X \cdot 1 = X$	2b. $X + 0 = X$
3a. $X \cdot X = X$	3b. $X + X = X$
4a. $X \cdot \bar{X} = 0$	4b. $X + \bar{X} = 1$
5. $\bar{\bar{X}} = X$	
6a. $X \cdot Y = Y \cdot X$	6b. $X + Y = Y + X$
7a. $X(YZ) = (XY)Z = (XZ)Y = XYZ$	
7b. $X + (Y + Z) = (X + Y) + Z = (X + Z) + Y = X + Y + Z$	
8a. $X \cdot (Y + Z) = XY + XZ$	8b. $X + YZ = (X + Y) \cdot (X + Z)$
9a. $\overline{X \cdot Y} = \bar{X} + \bar{Y}$	9b. $\overline{X + Y} = \bar{X} \cdot \bar{Y}$
10a. $X \cdot (X + Y) = X$	10b. $X + XY = X$
11a. $(X + Y) \cdot (X + \bar{Y}) = X$	11b. $XY + X\bar{Y} = X$
12a. $(X + \bar{Y}) \cdot Y = XY$	12b. $X\bar{Y} + Y = X + Y$
13a. $(X + Y) \cdot (\bar{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\bar{X} + Z)$	
13b. $XY + \bar{X}Z + YZ = XY + \bar{X}Z$	
14a. $X \oplus Y = (X + \bar{Y}) \cdot (\bar{X} + Y)$	14b. $X \oplus Y = \bar{X}Y + X\bar{Y}$
15a. $X \odot Y = (X + Y) \cdot (\bar{X} \cdot \bar{Y})$	15b. $X \odot Y = \bar{X}\bar{Y} + XY$
15c. $X \odot Y = (X + Y) \cdot (\bar{X} + \bar{Y})$	

Annulment Law  
 Identity Law  
 Idempotent Law  
 Complement Law  
 Double Negation Law  
 Commutative Law  
 Associative Law  
 Associative Law  
 Distributive Law  
 de Morgan's Theorem   
 Absorption Law  
 Redundancy Law  
 Redundancy Law  
 Consensus Law  
 Consensus Law  
 XOR Gate  
 XNOR Gate  
 XNOR Gate

■ There are others.

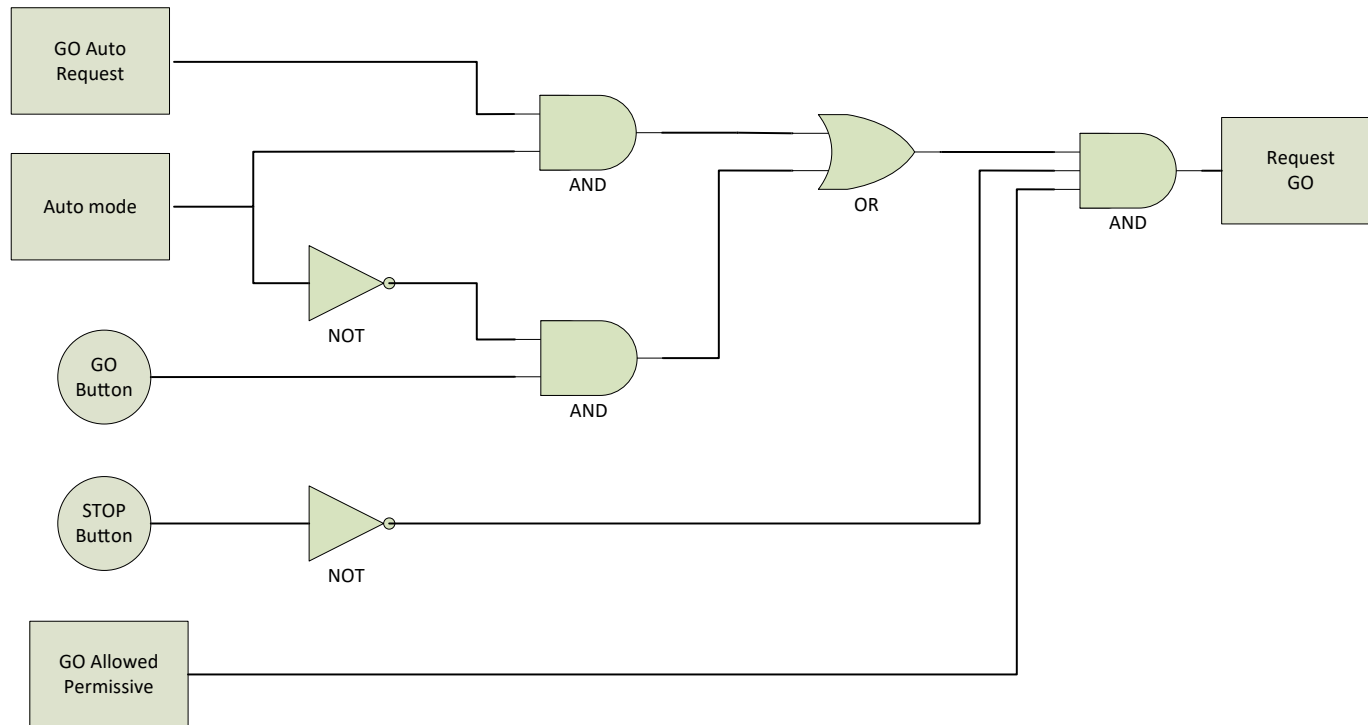
# Boolean Algebra – Simplification

- **Simplify to Product of Sums or Sum of Products**
  - Often for control systems, “product of sums” is better than “sum of products”.
    - Conditions AND permissives
- **There is much, much, more about this online.**

# Converting Numeric To Boolean

- The output of a comparison of two numeric values is a boolean.
- Examples:
  - Equal – True when  $A = B$
  - Not Equal – True when  $A \neq B$
  - Greater or Equal – True when  $A \geq B$
  - Less or Equal – True when  $B \leq A$
  - In Range – True when analog  $C \geq A$  and  $C \leq B$ . For the output to ever be true  $B$  must be  $> A$ .
- These functions often have a deadband value to reduce chatter.

# Sample 4.1



$$\text{GO\_Request} = \text{GO\_Perm} \cdot \text{GO\_allowed} \cdot \neg \text{STOP\_Button} \cdot (\text{GO\_Auto} \cdot \text{Auto\_mode} + \text{GO\_Button} \cdot \neg \text{Auto\_Mode})$$

# Exercise 4.1 – Ball shooter size detector

- Floor contains 3 sizes of balls. Only the middle size can be shot correctly.
  - The small balls have 80% the diameter of the middle ball
  - The large balls have 120% the diameter of the middle ball.
- The robot picks up any size balls and they roll along a belt. The belt is always on. When the balls reach the “front” sensor either “reject” the ball back onto the floor, or “shoot” the ball.
- Boolean Inputs (sensors):

1) Front of ball	2) 97% diameter of middle ball
3) 103% diameter of middle ball	4) Robot enabled (simulate with a sensor)
- Boolean Outputs (actuators):

1) Reject ball	2) Shoot ball
----------------	---------------
- When a ball is picked up it first encounters the 103% sensor, then the 97% sensor, then the front of ball sensor. Based on the size of the ball, multiple sensors may indicate true at the same time. When the front of ball sensor is TRUE either reject or shoot the ball based on its size. If the robot is not enabled, always reject the ball. Draw the logic and write the equation.

# Robot Training 01

- **Complete Robot Training 01 presentation**

# Exercise 4.2 – Ball shooter size detector

- **Implement the solution to 4.1 on a robot.**
- **The limit switch inputs use:**
  - Front of ball limit switch – DIO 0
  - 97% of ball limit switch – DIO 1
  - 103% of ball limit switch – DIO 2
  - Robot enabled limit switch – DIO 3
- **The DIO outputs (digital outputs) use:**
  - Shoot ball – DIO 8
  - Ball Detected – DIO 7
- **Use the robot project “put-name-here”. The only VI that need to be modified are in the “BallDetectShoot” sub-directory. They are:**
  - BallDetectShoot\_Open                      - One time initialization goes here
  - BallDetectShoot\_Execute                - Code to periodically execute goes here