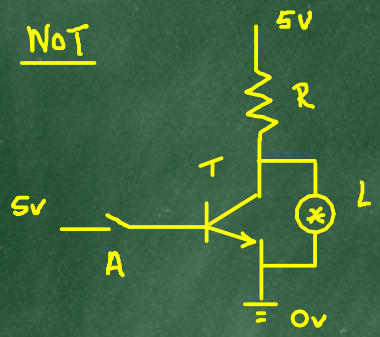
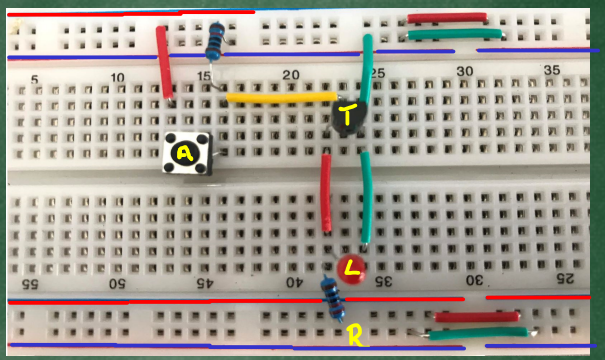
# Implementing AND, OR and NOT using discrete transistors

## NOT

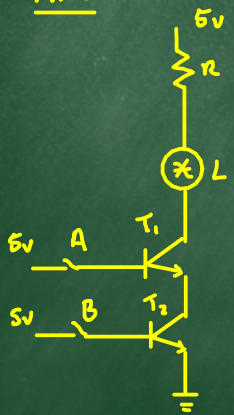



| A | L |
|---|---|
| 0 | 1 |
| 1 | 0 |

Input Output
0 → 0v, 0 → off
1 → 5v, 1 → on

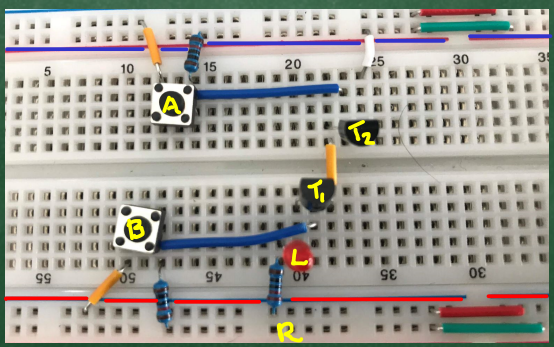



## AND

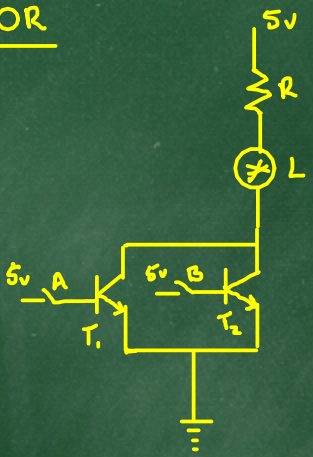



| A | B | L |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Input Output
0 → 0v, 0 → off
1 → 5v, 1 → on

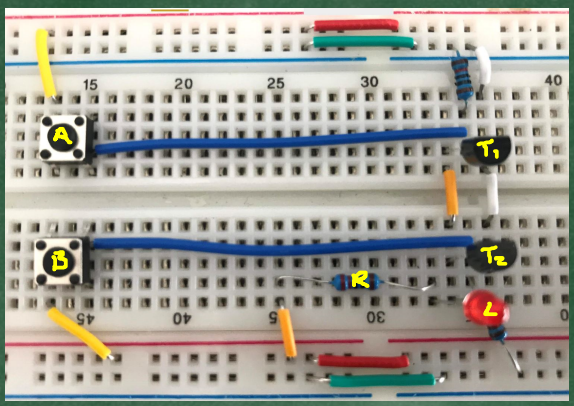


# OR




| A | B | L |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Input output
0 → 0v, 0 → off
1 → 5v, 1 → on



In addition to the fundamental set of AND, OR and NOT, it is common to consider other gates (whose functionality is needed often and whose construction is a combination of AND, OR, NOT).

## 1. Exclusive OR (XOR, EXOR)

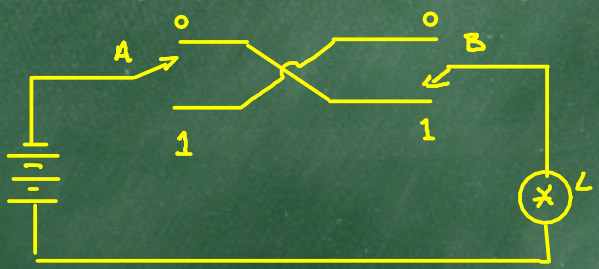
| A | B | A XOR B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| * 1 | 1 | 0 |

Algebraically: $A \oplus B$

In Circuit Symbol:


this case distinguishes OR and XOR

### XOR as an Electrical Circuit

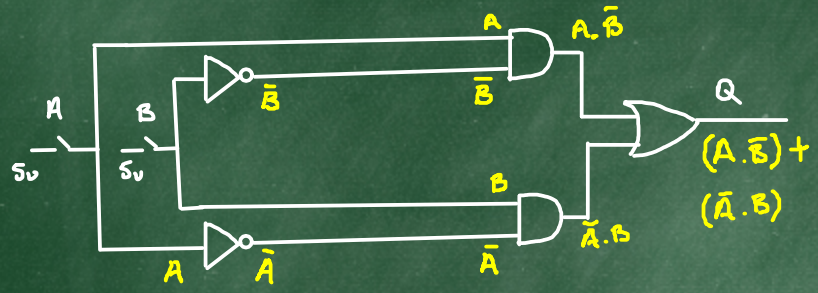



| A | B | L |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Where in your home do you have this?

# Implementing XOR with AND, OR, NOT

Consider the following:




| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Switch open – 0
Switch closed – 1

if we try all 4 options for A, B and trace them through the circuit we will find the output Q.

| A | B | $\bar{A}$ | $\bar{B}$ | $A.\bar{B}$ | $\bar{A}.B$ | $A.\bar{B}+\bar{A}B=Q$ | $=A\oplus B$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | |

This tabulation method is called **Perfect Induction**.

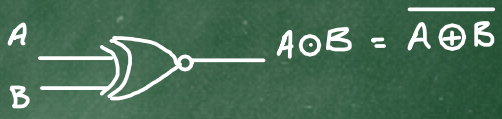
We will use this powerful technique again later.

## 2. XNOR – NOT of Exclusive OR

| A | B | A XNOR B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Algebraically: $A \odot B$

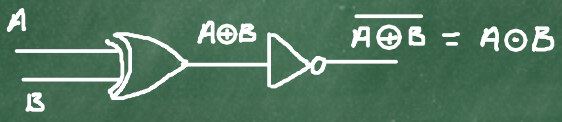
In Circuit Symbol:




XNOR is also called Coin – for Coincidence.

That is, the output is TRUE (1) if both Inputs Coincide and is False (0) otherwise.

Equivalent Circuit:



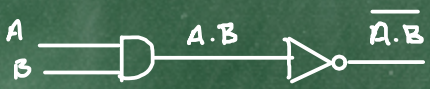
## 3. NAND (NOT OF AND)

| A | B | A NAND B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Algebraically : $\overline{A.B}$

In Circuit Symbol :


Equivalent Circuit



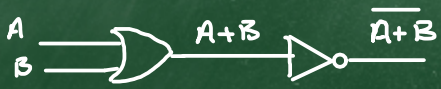

## 4. NOR (NOT of OR)

| A | B | A NOR B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Algebraically : $\overline{A+B}$

In Circuit Symbol :


Equivalent Circuit



# Small – Scale Integration (74xx Series)

## 7408

Quad, 2-Input AND Gate