1.0. LOGIC GATES

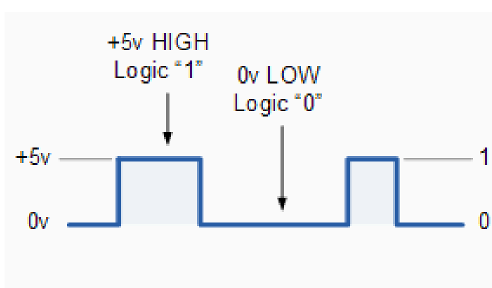
Objective:

- Getting started with Logisim
- To become familiar with the operation of Basic Gates (AND, OR, NOT, NAND, NOR, XOR, XNOR).
- To become familiar with how to determine the truth tables for logic gates.

1.1. Introduction

Logic deals with only two normal conditions: logic TURE or logic FALSE. In Boolean logic, TRUE is often represented by the term HIGH or the number 1 and FALSE is represented by the term LOW or the number 0. HIGH and LOW (1 or 0) are logic terms; they do not indicate whether the voltage is higher or lower. In positive logic the more positive voltage is TRUE and the less positive voltage is FALSE i.e +2.5V = HIGH and +0.5V = LOW. With the negative logic this definition is reversed.

The basic logic gates and their symbols are summarized in the following pages. The truth table with all possible input combination is given and the output is left empty to you as an exercise. All possible combination of inputs involve counting in binary from 0 to 2^n – 1 where n is the number of inputs.



In this experiment you will look at the truth tables for several arrangements of simple gates.

1.2. Gates

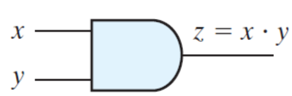
In this section, we will use logicsim to simulate the 7 gates ad get their truth tables

- Start the logicsim
- Connect the following gates and simulate to prove the truth tables

1.2.1. AND Gate

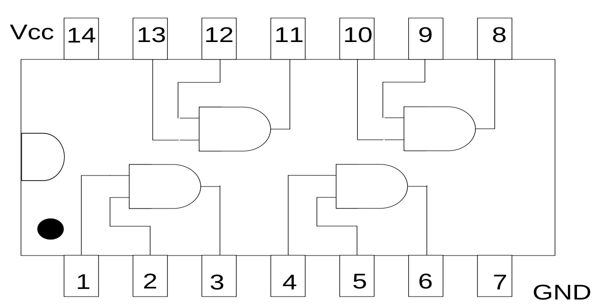
A Logic AND Gate is a type of digital logic gate that has an output which is normally at logic level "0" and only goes "HIGH" to a logic level "1" when ALL of its inputs are at logic level "1". The output of a Logic AND Gate only returns "LOW" again when ANY of its inputs are at a logic level "0". The logic or Boolean expression given for a logic AND gate is that for Logical Multiplication which is denoted by a single dot or full stop symbol, (.) giving us the Boolean expression of:

$$z = x . y$$



x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Example of an AND Gate IC: 74LS08

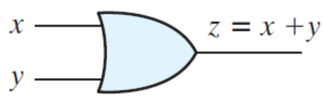


Quad 2-input AND gate 7408

1.2.2. OR Gate

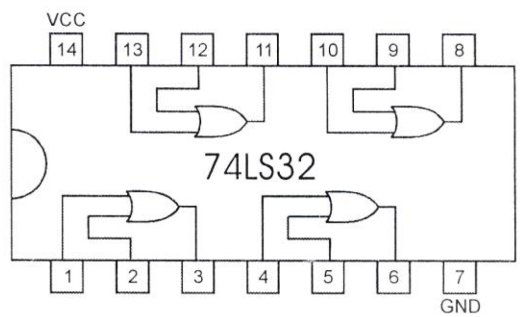
The two inputs marked as x and y provides an output z. The OR gate has an output of 1 when either A or B or both are 1.

$$z = x + y$$



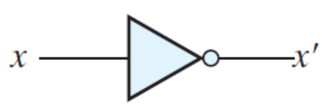
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Example of an OR Gate IC: 74LS32



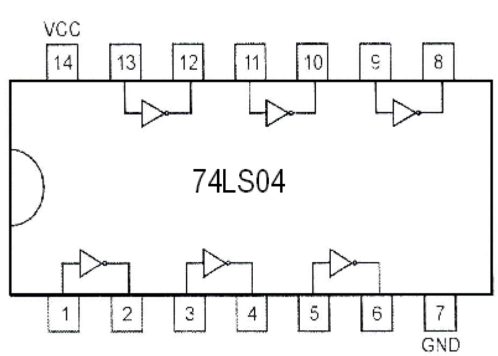
1.2.3. NOT Gate

It is so called because its output is NOT same as its inputs. It is also called an inverter because it inverts the input signal. It has one input and one output. All it does to invert (or complement) the inputs.



x	x'
0	1
1	0

Example of an NOT Gate IC: 74LS04

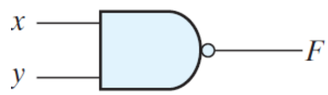


1.2.4. NAND Gate

It is in fact a NOT-AND gate .It can be obtained by connecting a NOT gate in the output of an AND gate. Its output is given by the Boolean equation.

$$Z = (x \cdot y)'$$

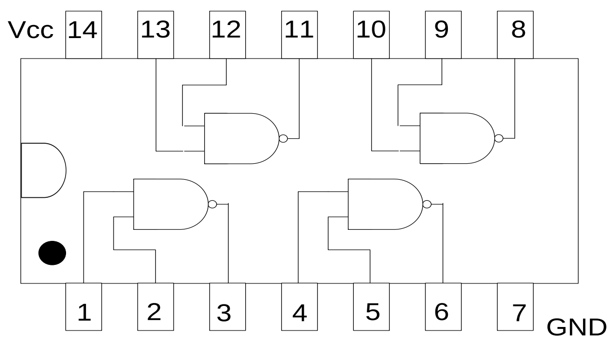
This gate gives an output high if either A or B or both are 0.



$$F = (xy)'$$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

Example of an NAND Gate IC : 74LS00



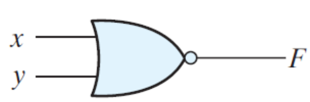
Quad 2-input NAND gate 7400

1.2.5. NOR Gate

It is a NOT-OR gate. It can be obtained by connecting a NOT gate in the output of an OR gate. Its output is given by the Boolean equation.

$$z = (x + y)'$$

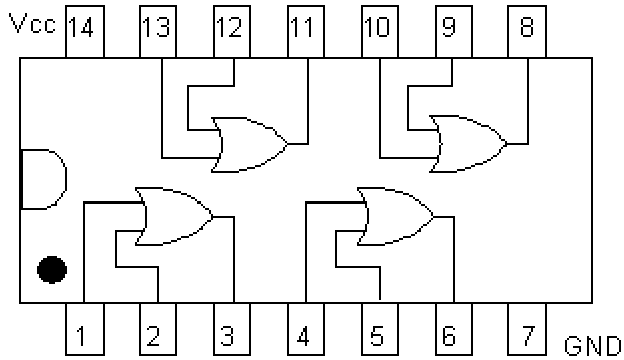
It gives an output when it's both inputs are 0.



$$F = (x + y)'$$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

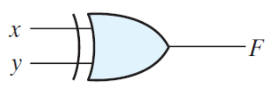
Example of an NOR Gate IC: 74LS02



Quad 2-input OR gate 7432

1.2.6. XOR Gate

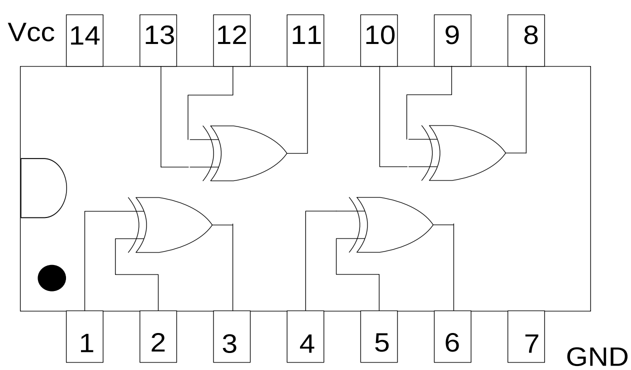
It is the gate which gives an output 1 when its inputs are not same (or exclusive) and an output 0 when its inputs are same.



$$F = xy' + x'y$$
$$= x \oplus y$$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

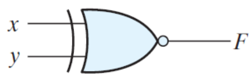
Example of an XOR Gate IC: 74LS86



Quad 2-input XOR gate 7486

1.2.7. X-NOR Gate

It is the gate which gives an output 0 when its inputs are not same (or exclusive) and an output 1 when its inputs are same.



$$F = xy + x'y' = (x \oplus y)'$$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

1.3. Review Questions

- Why NAND & NOR gates are called universal gates?
- Realize the EX – OR gates using minimum number of NAND gates.
- Give the truth table for EX-NOR and realize using NAND gates?