Indian Institute of Technology Ropar

Department of Electrical Engineering

EE204 : Digital Circuits Laboratory

Classroom - Analog and Digital Circuits Lab

Introduction:

Any digital logic can be realized using three basic operations: AND, OR and NOT (inversion). For this reason, circuits that can implement these functions are called basic logic gates. These basic logic gates are available in standard TTL and CMOS ICs and form the functional building blocks for complex combinational and sequential circuits. If we carefully examine the truth tables of these logic gates, we observe that they can be realized using basic switches. These switches can be elementary toggle switches (SPDT) or discrete solid-state devices (p-n junction diodes and transistors). In our current lab exercise, we shall be using switches to implement these logic functions.

Aim:

Implementation of basic logic gates using switches, p-n junction diodes and bipolar junction transistors.

Theory:

The basic logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR. Figures below show the circuit symbols, Boolean function, and truth. It is seen from the Fig that each gate has one or two binary inputs, A and B, and one binary output, C. The small circle on the output of the circuit symbols designates the logic complement. The AND, OR, NAND, and NOR gates can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative. These basic logic gates are implemented as small-scale integrated circuits (SSICs) or as part of more complex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic operation, but also the specific logic circuit family to which they belong. Each logic family has its own basic electronic circuit upon which more complex digital circuits and functions are developed. The following logic families are the most frequently used.

1. Transistor-transistor logic (TTL)
2. Emitter-coupled logic (ECL)
3. Metal-oxide semiconductor (MOS)
4. Complementary metal-oxide semiconductor (CMOS)

TTL and ECL are based upon bipolar transistors. TTL has a popularity among logic families & ECL is used only in systems requiring high-speed operation. MOS and CMOS, are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

Pre-Lab quiz:

Q. Why are NAND & NOR gates called universal gates?

Q. Compare TTL logic family with CMOS family?

Q. Which logic family is fastest and which has low power dissipation?

Components:

SPDT switches, p-n junction diodes, BJT, LEDs, power supply and standard experimental setup.

Procedure:

The truth tables and output Boolean functions for the basic gates are given below. Implementation of these gates using switches, diodes and BJTs is explained in Fig. 1. Using the given information, implement the circuits on the breadboard. For standard inputs (A & B), obtain the corresponding output (Y) and verify the truth tables for these logic gates.

a. *AND gate*: The truth table and the output expression for a two input AND gate are given below:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = A \,.\, B$$

b. *OR gate*: The truth table and the output expression for a two input OR gate are given below:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

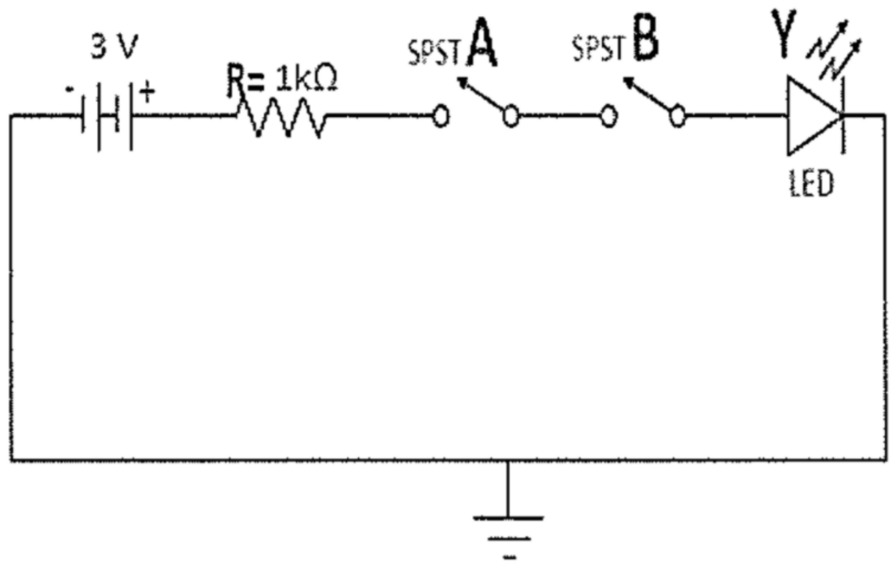
$$Y = A + B$$

c. *NOT gate*: The truth table and the output expression for a NOT gate (inverter) are given below:

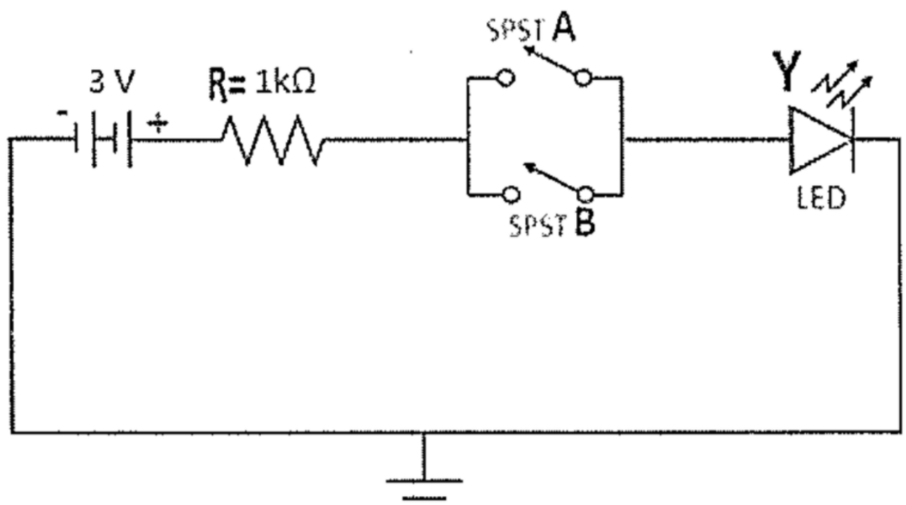
A	Y
0	1
1	0

$$Y = \bar{A}$$

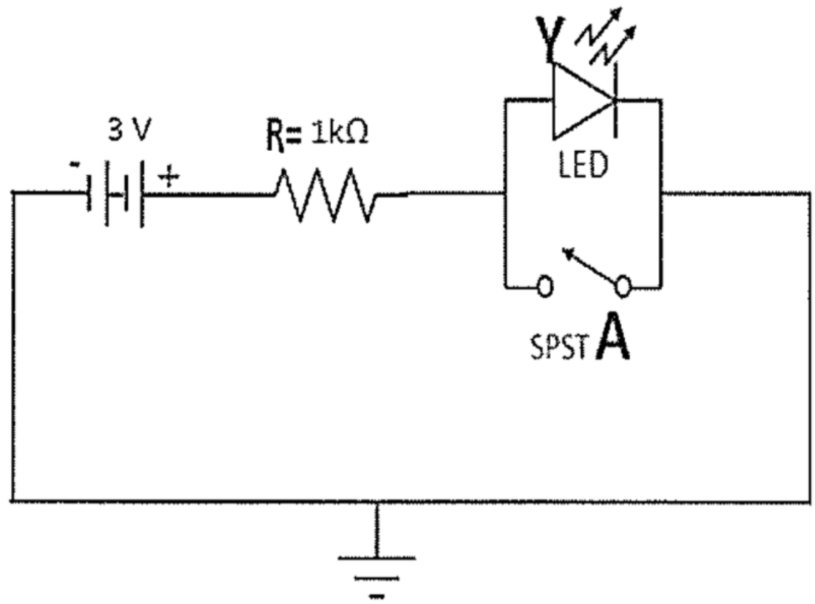
AND gate using switches



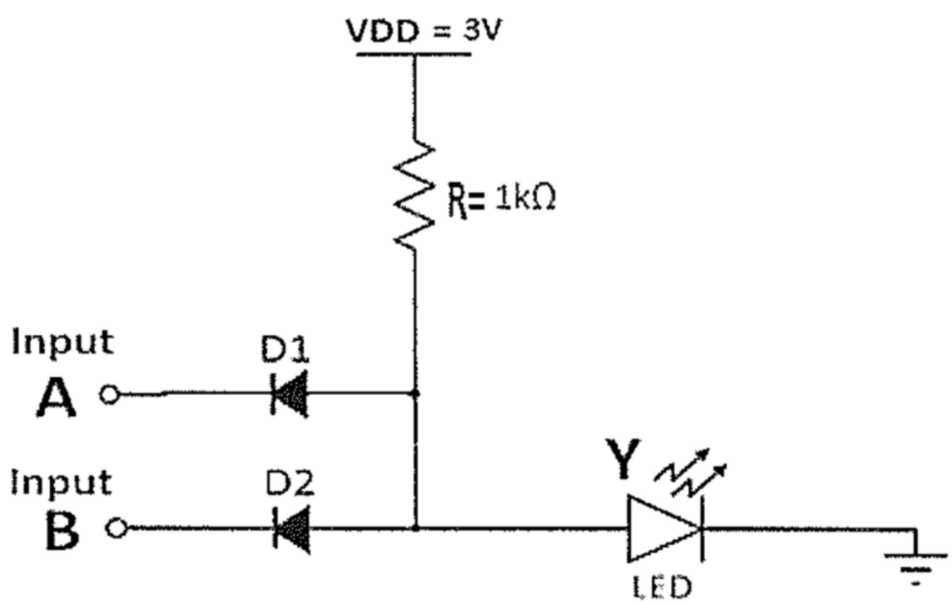
OR gate using switches



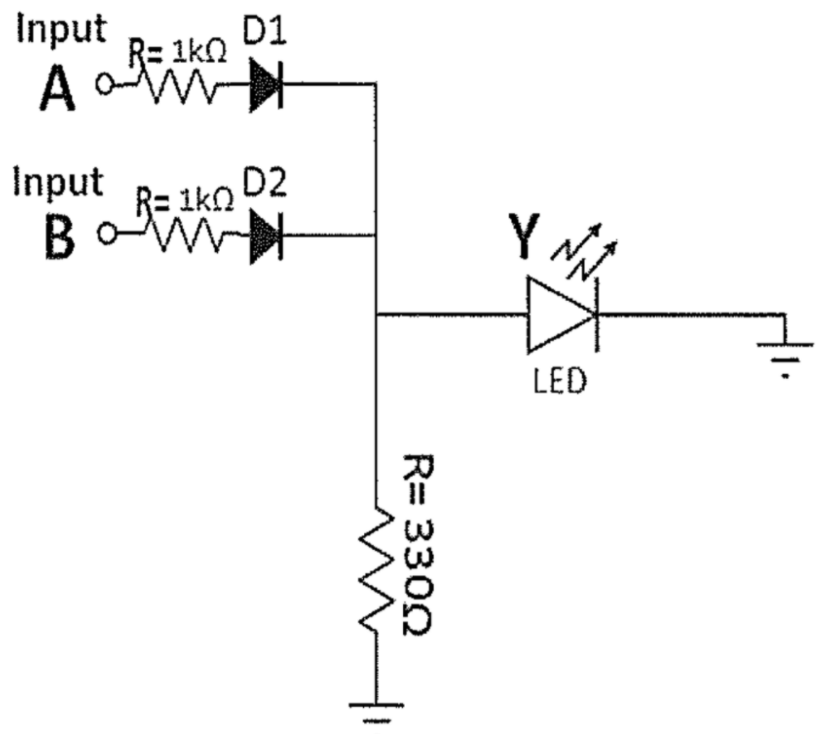
NOT gate using a switch



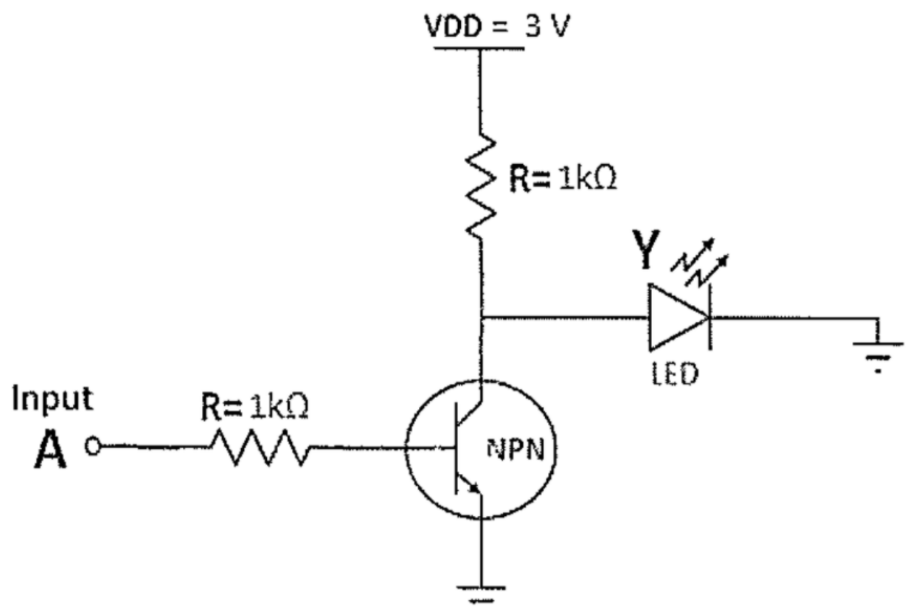
AND gate using diodes



OR gate using diodes



NOT gate using a BJT



Post-Lab quiz:

Q. Can we design a NOT gate using diodes? If not, give reasons.

Q. How would you implement AND and OR logic using BJTs?

Q. Can we simulate the above logic circuits using a circuit simulator (eg. SPICE)? Give its netlist (program).

Suggested Readings: M. Morris Mano, "Digital Logic and Computer Design", Pearson Prentice Hall, 2008.