MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
(Approved by AICTE , Affiliated to Pondicherry University and Accredited by NBA)
Kalitheerthalkuppam, Puducherry - 605107.

Department of Electronics and Communication Engineering

Virtual Lab

Variation of magnetic field with the distance along the axis of a current carrying coil and estimate the radius of the coil

Aim:

To determine the variation of magnetic field with the distance along the axis of a current carrying coil and estimate the radius of the coil.

Apparatus Required:

1. Circular coil (N turns)
2. Power supply (DC)
3. Ammeter
4. Magnetic field sensor / Gauss meter / compass with a ruler
5. Meter scale or vernier caliper
6. Rheostat (optional)

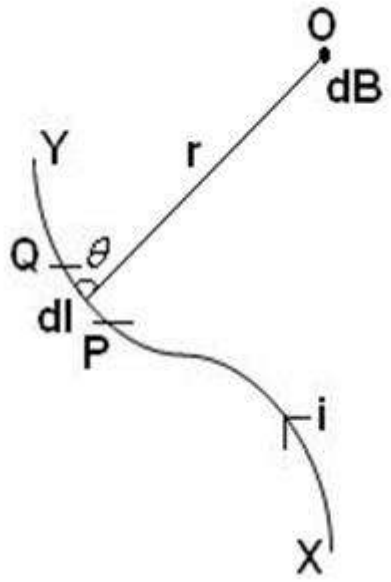
Theory:

The magnetic field created by a current carrying coil can be determined using Biot-Savart's law. According to the law, the magnetic field at a point P due to a small current element dl at a distance r from P is given by:

$B = (\mu 0 * I * dl * \sin(\theta)) / (4 * \pi * r^2)$

Where:
B is the magnetic field,
μ_0 is the permeability of free space,

I is the current in the wire,
dl is a segment of the wire,
θ is the angle between the direction of the current and the direction to the point where the magnetic field is being measured,
r is the distance from the wire to the point where the magnetic field is being measured.



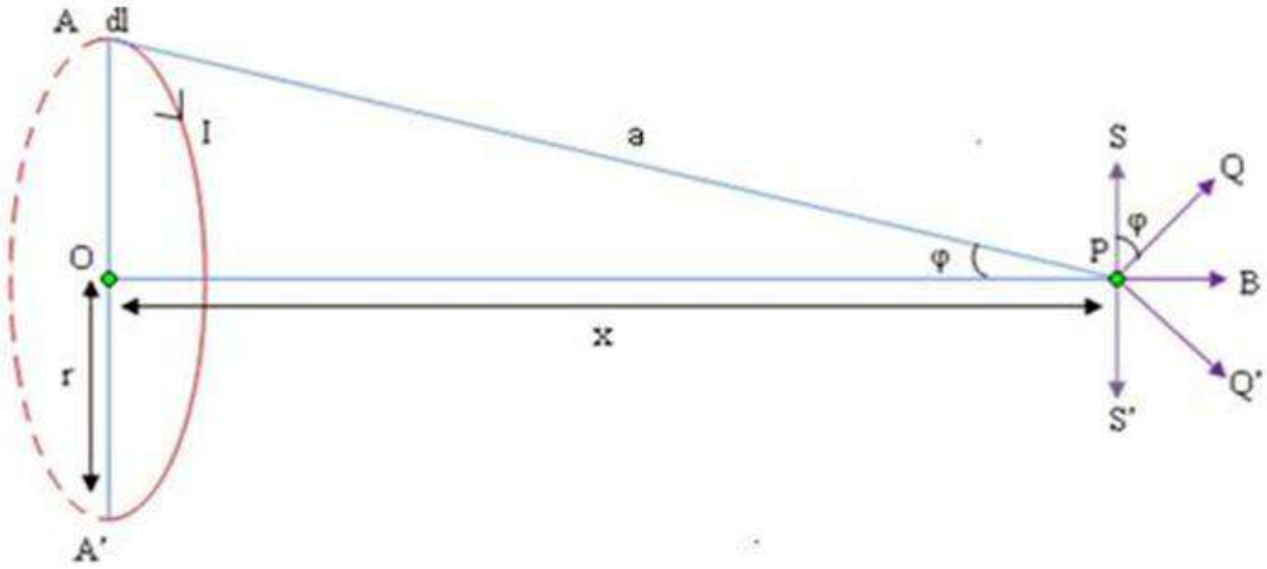
The magnetic field at a point along the axis of a current carrying coil decreases with increasing distance from the coil. This can be seen from the fact that the magnetic field is inversely proportional to the distance r.

The radius of the coil can be estimated by measuring the magnetic field at various points along the axis of the coil and using the above equation. By comparing the measured magnetic field with the calculated magnetic field using the Biot-Savart law, the radius of the coil can be estimated. The magnetic field along the axis of the coil can be obtained by summing the contributions of all the current elements in the coil.

The radius of the coil can be estimated using the formula for the magnetic field B, the number of turns N, and the current I in the coil:

$B = (\mu_0 * N * I) / (2 * R)$

Where R is the radius of the coil. Solving for R and substituting known values, the radius of the coil can be calculated.



Procedure:

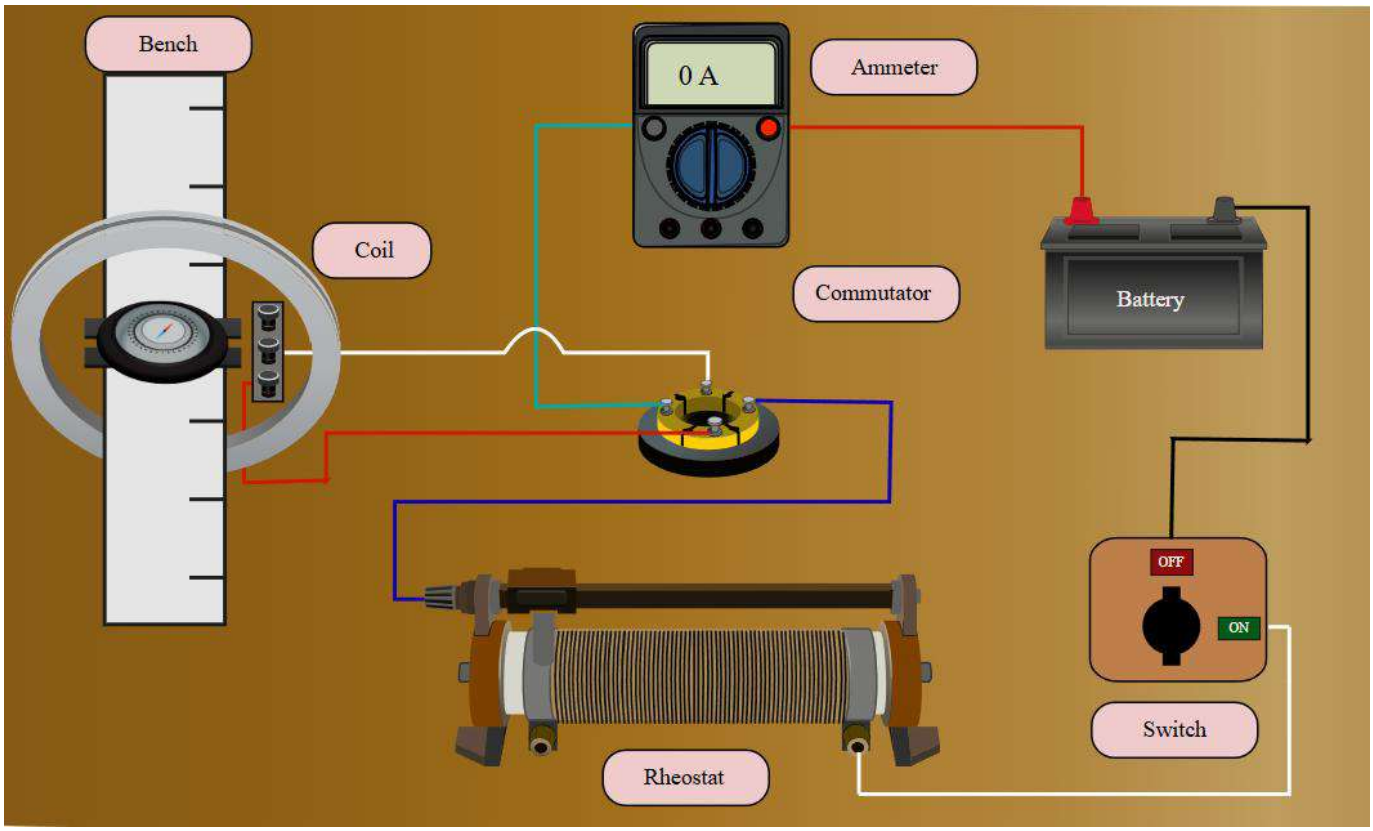
To determine the variation of magnetic field with the distance along the axis of a current carrying coil, follow these steps:

1. Obtain a current carrying coil and a Gaussmeter or a Hall effect sensor to measure magnetic field.
2. Place the Gaussmeter or Hall effect sensor at a fixed distance from the center of the coil and measure the magnetic field.
3. Repeat step 2 for different distances along the axis of the coil.
4. Plot the magnetic field vs distance graph to see the variation.

To estimate the radius of the coil, follow these steps:

1. Measure the magnetic field at the center of the coil.
2. Measure the current flowing through the coil and the number of turns in the coil.
3. Use the formula for magnetic field due to a solenoid to calculate the radius of the coil.
4. Compare the calculated radius with the actual radius to determine accuracy.

Simulation:



Tabulation:

S. no	Distance from center	Left side				Right side				mean θ degree	tanθ	$B_x(T)$	$B_o(T)$
		direct(radian)		reversed(radian)		direct(radian)		reversed(radian)					
		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8				
1	0	0	0	0.941	0	0.941	0.572	0	0	17.575480365638	0.317	1.109	1.109
2	5	0.9596669873457683	0.96	1.7	0.96	1.7	1.834	0.96	0.96	71.86084642683201	3.052	10.682	10.682
3	5	1.2338537380230532	1.234	1.617	1.234	1.617	1.26	1.234	1.234	76.37422656669095	4.125	14.438	14.438
4	10	0.1786222535620271	0.179	1.191	0.179	1.191	0.673	0.179	0.179	28.287085725006655	0.538	1.883	1.883
5	10	0.6254612799544339	0.625	1.318	0.625	1.318	1.431	0.625	0.625	51.51220945658139	1.258	4.403	4.403
6	10	0.14344518044516205	0.143	0.298	0.143	0.298	0.554	0.143	0.143	13.360266969066009	0.238	0.833	0.833

Pretest:

Unit of magnetic permeability (μ) ?

- a: unitless quantity Explanation
- b: henry per meter squared Explanation
- c: Watt per meter Explanation
- d: henry per meter Explanation

If the radius of a circular current carrying coil is doubled then the magnetic field at the center of the coil becomes- (all other factors remains s

- a: two times Explanation
- b: half Explanation
- c: four times Explanation
- d: remains same Explanation

What fundamental principle describes the magnetic field produced by a current-carrying coil?

- a: Faraday's Law Explanation
- b: Ampère's Law Explanation
- c: Ohm's Law Explanation
- d: Biot-Savart's Law Explanation

Submit Quiz

Score: 3 out of 3

Posttest:

1. What does 'dl' represent in the context of Biot-Savart's Law?
- a: Current in the wire Explanation
- b: A segment of the wire Explanation
- c: Permeability of free space Explanation
- d: Distance from the wire to the point where the magnetic field is being measured Explanation

2. How does the magnetic field along the axis of a current-carrying coil change with increasing distance from the coil?
- a: It remains constant Explanation
- b: It decreases linearly Explanation
- c: It decreases with increasing distance Explanation
- d: It increases with increasing distance Explanation

3. How can the radius of a coil be estimated without direct measurement?
- a: By counting the number of turns in the coil Explanation
- b: By measuring the current passing through the coil Explanation
- c: By comparing measured magnetic fields with theoretical calculations using Biot-Savart's Law Explanation
- d: By measuring the voltage across the coil Explanation

4. In the formula involving the magnetic field, number of turns, current, and radius of the coil, what physical quantity is used to estimate the coil's radius?
- a: Resistance Explanation
- b: Permeability of free space Explanation
- c: Magnetic field strength Explanation
- d: Inductance Explanation

5. What other methods can be used to determine the variation of magnetic field with distance along the axis of a current carrying coil?
- a: Using a compass Explanation
- b: Using a voltmeter Explanation
- c: Using an oscilloscope Explanation
- d: Using an oscillograph Explanation

Submit Quiz

Score: 5 out of 5

Measurement of high resistance by the method of leakage of a condenser

Aim:

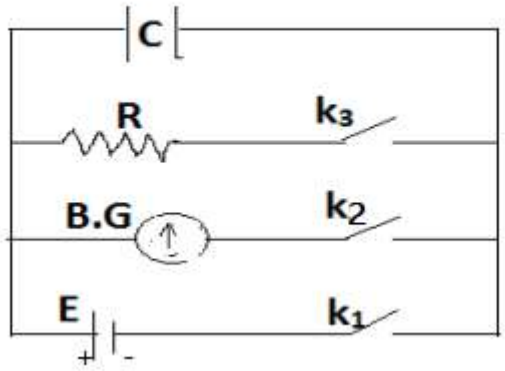
Measurement of high resistance by the method of leakage of a condenser

Apparatus Required:

1. Ballistic galvanometer
2. High resistance
3. Capacitor (e.g. 2 µF)
4. Key (2-way)
5. Battery (around 6V)
6. Stopwatch
7. High-value resistor (to be determined)
8. Charging and discharging circuit

Theory:

Consider the circuit shown in the figure a condenser C, resistance R, Ballastic Galvanomenter B.G. and cell of emf E is connected in as shown in the circuit. Keeping K_2 and K_3 open, the capacitor is charged by depressing the key K_1. K_1 is then opened and at once K_3 is closed. The capacitor discharges through the galvanometer, θ_0 is proportional to Q_0. The capacitor is again charged, the throw which records a throw to the maximum value keeping K_2 and K_3 open and K_1 closed. K_1 is then opened and K_2 is closed for a known time t. Some of the charge leaks through R. K_2 is opened and at once K_3 is closed. The charge Q remaining on the capacitor is noted by noting the value of θ (deflection produced in the Ballastic Galvanometer).



If a charged condenser of capacity C (having self leakage resistance 'r') is discharged through any high resistance R in time't' then

$$\log_{10}\frac{\theta_0}{\theta} = \frac{t}{2.303RC} \qquad \text{-------- (1)}$$

the graph between $\log_{10}\frac{\theta_0}{\theta}$ and t will be straight line.

where θ_0 = the deflection corresponding to the first throw of the galvanometer θ = the deflection of the galvanometer when the capacitor is discharged through resistor R. The relation (1) can also be written as

$$\log_{10}\frac{\theta_0}{\alpha} = \frac{t}{2.303R'C} \qquad \text{-------- (2)}$$

Where α = the deflection of the galvanometer when the capacitor is discharged through resistor R'
and R' is the resistance of the equivalent parallel resistance of the galvanometer and self leakage resistance r of the condenser.

$$\frac{1}{R'} = \frac{1}{r} + \frac{1}{R} \qquad \text{---------- (3)}$$

Procedure:

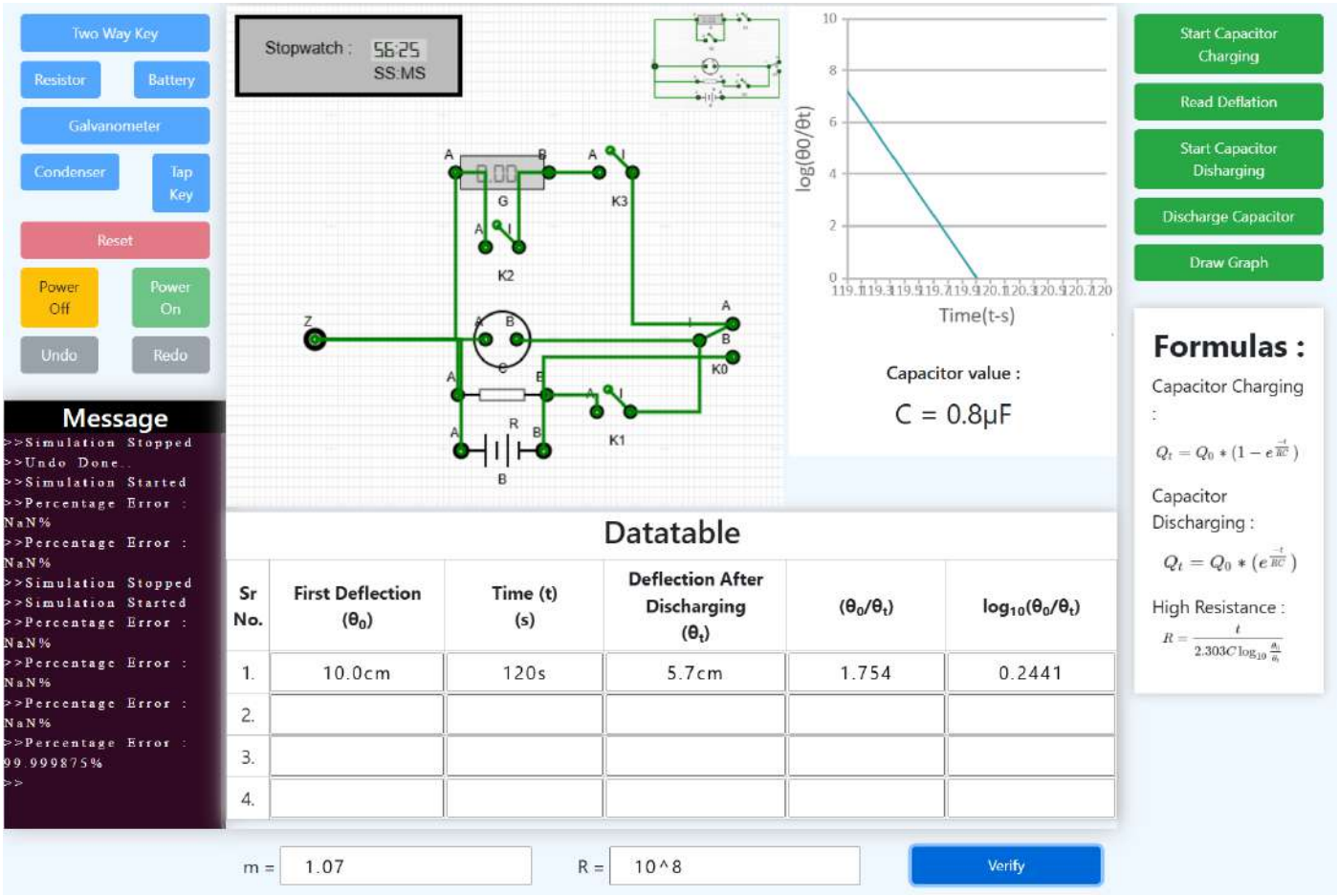
1. To draw circuit diagram on the drawing board follow steps 2 to 7
2. Click on the "To way key" button and then click on the drawing box.
3. Click on the "Resistor" button and then click on the drawing box.
4. Click on the "Battery" button and then click on the drawing box.
5. Click on the "Galvanometer" button and then click on the drawing box.
6. Click on the "Condenser" button and then click on the drawing box. Proper connections will be highlighted as green
7. Click on the points to drawing board to complete the circuit.
8. To start the simulation follow points 9 to 13
9. Click on the "Power On" button.
10. Click on the "Undo/Redo" button in case of wrong connection.
11. Click on the "Start Charging Capacitor" button.
12. Read the time in the Stop Watch.
13. Click "Read Deflection" button.
14. Enter the value of First Deflection and the time in the data table.
15. Click on the "Start Discharging Capacitor" button.
16. Enter the value of Deflection after discharging"
17. Repeat the experiment to complete the data table.
18. Click "Draw Graph" button.
19. Calculate the slope of the graph and enter the value m.
20. Calculate the slope of the graph and enter the value R.

Result:

The high resistance RRR measured by the method of leakage of a condenser is approximately:

$$\boxed{1.07 \times 10^8\ \Omega\ (\text{or } 107\ \text{M}\Omega)}$$

Simulation & Tabulation:



Datatable

Sr No.	First Deflection (θ_0)	Time (t) (s)	Deflection After Discharging (θ_t)	(θ_0/θ_t)	$\log_{10}(\theta_0/\theta_t)$
1.	10.0cm	120s	5.7cm	1.754	0.2441
2.					
3.					
4.					

Pretest:

1. The plot of $\log_{10}(\theta_0/\theta)$ and t for the circuit in the experiment is:
- a: Straight line
- b: parabolic
- c: hyperbolic
- d: eliptical

2. The capacity of the condenser connected to the circuit is of the order of
- a: μF
- b: pF
- c: nF
- d: none of these

3. Unit of RC is:
- a: m
- b: L
- c: Sec
- d: none of these

4. The self leakage resistance of the condenser and the resistance of the Ballistic Galvanometer are in:
- a: series
- b: parallel
- c: both
- d: none of these

Submit Quiz

4 out of 4

Posttest:

1. Slope of the graph between $\log_{10}\frac{\theta_0}{\alpha}$ and t is:

- [x] a: Always positive
- [ ] b: Always negative
- [ ] c: Positive then negative
- [ ] d: Negative then positive

2. The order of the unknown resistance is:.

- [x] a: MΩ
- [ ] b: Ω
- [ ] c: μΩ
- [ ] d: KΩ

3. The deflection in the Ballastic Galvanometer is directly proportional to:

- [ ] a: Charge present in the Battery
- [x] b: Charge present in the condenser
- [ ] c: Current in the circuit
- [ ] d: NOT

4. Charge grows in the condenser:

- [ ] a: Linearly
- [x] b: exponentially
- [ ] c: logarithmically
- [ ] d: NOT

Submit Quiz

4 out of 4

VIRTUAL LAB

DESIGN OF DIGITAL CIRCUITS USING VERILOG

AIM:

To design, simulate, and analyze digital circuits using the Verilog hardware description language, and to understand its structure, syntax, and practical applications in digital system design.

APPARATUS REQUIRED:

- EDA Tool: Xilinx Vivado / ModelSim / Icarus Verilog
- Verilog-compatible simulator (ModelSim / iVerilog)
- PC / Laptop with Windows or Linux OS
- GTKWave (for viewing simulation waveforms)
- Verilog source files (.v) and testbench files (_tb.v)

THEORY:

Digital circuits are built using basic logic gates such as AND, OR, NAND, and NOR. These gates form the foundation of all digital systems, enabling complex operations by combining simple binary decisions.

Logic Gates and Their Functions

- AND Gate: Output is high (1) only if both inputs are high.
- OR Gate: Output is high if at least one input is high.
- NAND Gate: Output is low only if both inputs are high (inverse of AND).
- NOR Gate: Output is high only if both inputs are low (inverse of OR).

VERILOG IMPLEMENTATION:

Each gate can be implemented in Verilog using the assign statement:

TRUTH TABLE:

A	B	AND	OR	NAND	NOR
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	0

- The assign statement is used for combinational logic.
- Inputs are declared as A and B, output as Y**.**

Testbench and Simulation
To verify the logic gates, a testbench is used to apply all possible input combinations and observe the output:

- Declare A and B as registers, Y as a wire.
- Instantiate the gate module.
- Apply input combinations (00, 01, 10, 11) and observe Y.

Experiment Simulation Steps

- The simulation page divides the Verilog module and testbench into colored code blocks.
- Drag and drop the code blocks to arrange them in the correct order.
- Complete the code blocks by entering module names, selecting inputs/outputs, and filling in the assign statement.
- Click Validate to check your code and view the output truth table.

OBJECTIVE:

1. To learn the basic concepts of verilog programming.
2. To design multiplexers, counters etc. using verilog coding.

PROCEDURE:

<u>Steps to Complete the Code</u>

1. **Arrange the Code Blocks:**
    - Place the code block that defines the Verilog module name, inputs, and outputs first.
    - Next, add the code block that defines the module functionality (the assign block).
    - Finally, add the code block that ends the module.
2. **Drag and Drop:**
    - Drag and drop the code blocks to arrange them in the order mentioned above for both the Verilog module and the testbench.
3. **Complete the Code Blocks:**
    - For the Verilog module:
        - Enter a name for the module. The name should begin with an alphabet and can include alphanumeric characters and underscores (_). No spaces or special characters allowed.
        - Select the inputs as A and B, and the output as Y.
        - In the assign block, assign Y the value of the required logic expression:
            - For AND logic: Y = A & B;
            - For OR logic: Y = A | B;
            - For NOR logic: Y = ~(A | B);
            - For NAND logic: Y = ~(A & B);
        - Use the blocking assignment operator (=) for combinational logic.

SAMPLE VERILOG CODE:

```
// AND Gate
          module and_gate (
              input A,
              input B,
              output Y
          );
              assign Y = A & B;
          endmodule

          // OR Gate
          module or_gate (
              input A,
              input B,
              output Y
          );
              assign Y = A | B;
          endmodule

          // NAND Gate
          module nand_gate (
```

- For the Verilog testbench:
  - Enter a name for the testbench. The name should begin with an alphabet, include only alphanumeric characters and underscores, and must not match the module name.
  - Declare A and B as registers, and Y as a wire.
  - Instantiate the logic gate module by entering its name and selecting the arguments in the same order as in the module definition.
  - Define the input A and B waveforms in the initial block.

OBSERVATIONS:

After running the Verilog simulation, the following behavior is observed:

1. **AND Gate:** When both inputs A = 1 and B = 1 (HIGH), the output Y = 1 (HIGH). For all other input combinations, the output Y remains 0 (LOW), confirming the AND gate operation.
2. **OR Gate:** When both inputs A = 0 and B = 0 (LOW), the output Y = 0 (LOW). For all other input combinations, the output Y = 1 (HIGH), confirming the OR gate operation.
3. **NAND Gate:** When both inputs A = 1 and B = 1 (HIGH), the output Y = 0 (LOW). For all other input combinations, the output Y remains 1 (HIGH), confirming the NAND gate as an inverted AND operation.
4. **NOR Gate:** When both inputs A = 0 and B = 0 (LOW), the output Y = 1 (HIGH). For all other input combinations, the output Y = 0 (LOW), confirming the NOR gate as an inverted OR operation.
5. The output waveforms of all four gates correctly match their respective truth tables, validating the Verilog simulation results.
6. The simulation confirms that NAND and NOR gates produce complemented outputs compared to AND and OR gates respectively, which is consistent with universal gate behavior.

```
    input A,
    input B,
    output Y
);
    assign Y = ~(A & B);
 endmodule

// NOR Gate
module nor_gate (
    input A,
    input B,
    output Y
);
    assign Y = ~(A | B);
endmodule
```

OBJECTIVE:

1. To design a 2:1 multiplexer using pass transistor logic.
2. To implement the multiplexer circuit using SPICE netlist code.
3. To analyze the circuit behavior through transient simulation.

COMPONENTS REQUIRED:

- SPICE Simulator with NgSpice and PTM_45nm.txt model file
- NMOS transistors (pass transistor logic): W = 0.5µm, L = 45nm
- PMOS transistors (pass transistor logic): W = 1.0µm, L = 45nm
- Inverter subcircuit (to generate S')
- Load capacitor (cLoad): 50 fF
- Pulse voltage sources for A, B, and select S

PROCEDURE:

Circuit Connections:

- Transmission Gate 1 (TG1): Controlled by S' (NMOS gate) and S (PMOS gate). Input A passes to output Y when S = 0.
- Transmission Gate 2 (TG2): Controlled by S (NMOS gate) and S' (PMOS gate). Input B passes to output Y when S = 1.
- An inverter subcircuit generates S' from the select line S.
- PMOS bulk terminals are connected to VDD; NMOS bulk terminals to GND.
- A load capacitor (cLoad = 50 fF) is connected at the output node Y.

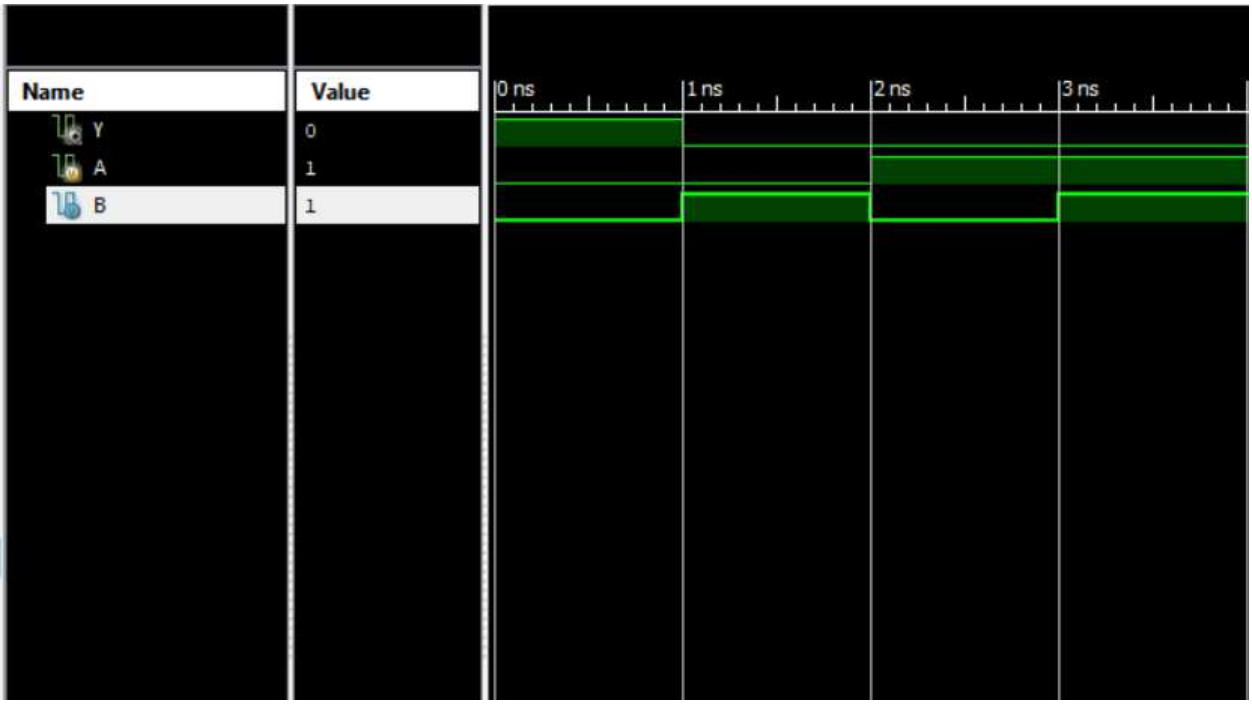
Steps to Perform the Simulation:

- Open the Practice tab of the virtual lab simulator.
- Arrange colored SPICE code blocks in order: Blue → Green → Yellow → Red → Teal
  → Orange → Purple → Gray.
- Blue Block: Include PTM_45nm.txt model file and declare parameters.
- Green Block: Define VDD and VSS/GND supply voltage sources.
- Yellow Block: Define the inverter subcircuit (.subckt inverter ...) with PMOS and NMOS.
- Red Block: Define the pass transistor (transmission gate) subcircuit using the inverter.
- Teal Block: Instantiate the 2:1 MUX subcircuit in the top-level netlist.
- Orange Block: Declare input waveforms for A, B, and S as pulse voltage sources.
- Purple Block: Add .tran analysis command and .plot output statement.
- Click 'Validate' — on success, simulation graphs appear in the Observations tab.
- Use 'Expand Waveform' for a larger view of input/output waveforms.

OUTPUT:

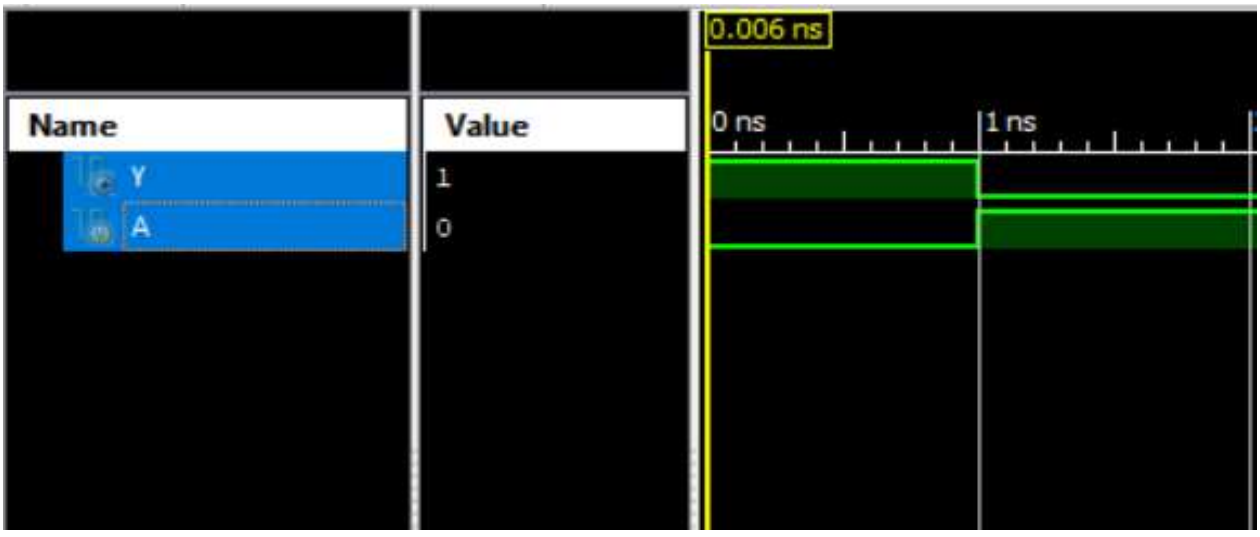
OR GATE:

NOR GATE:



NAND GATE:

NOT GATE:



RESULT:

The AND, OR, NAND, and NOR gates were successfully designed and simulated using Verilog HDL in the Virtual Lab environment. The simulation waveforms for both circuits confirmed that the output matches the expected truth table values for all input combinations. The Verilog data-flow modeling approach using 'assign' statements was validated, and the functional correctness of both circuits was established through exhaustive testbench simulation.