**Experiment No:** 1

**Experiment Name:** Implementation of CMOS NAND gate.

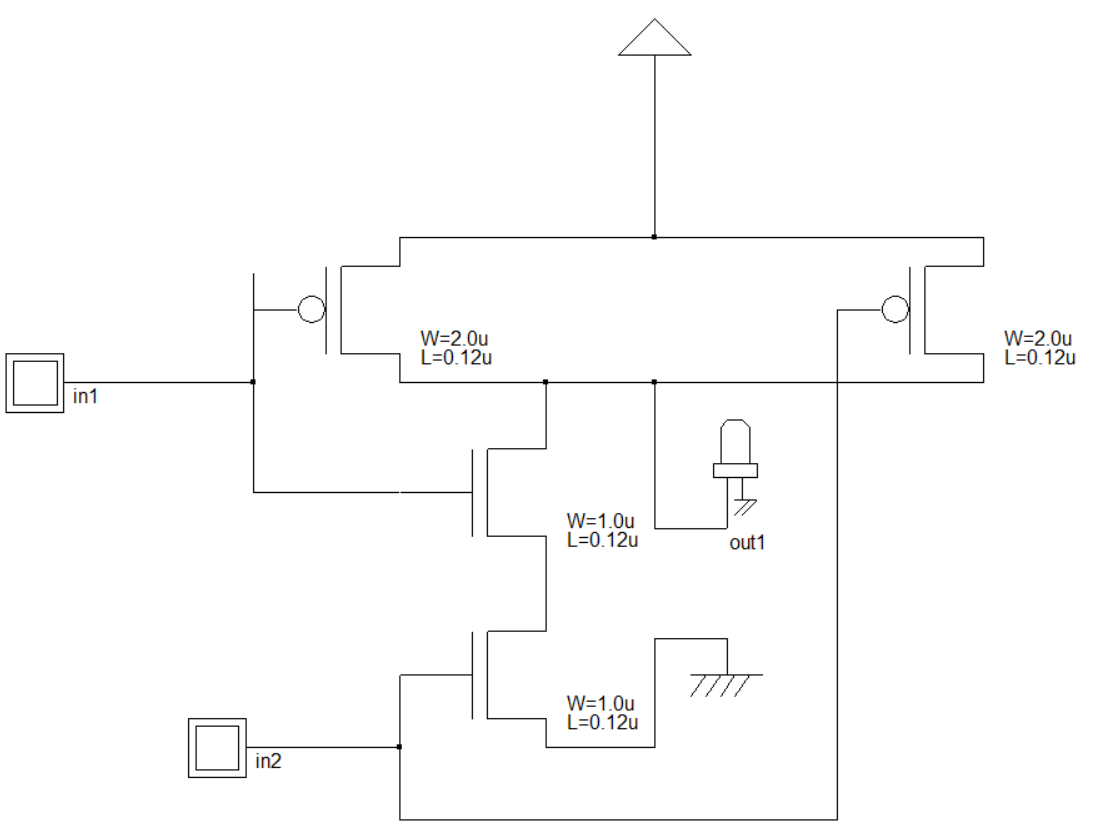
**Objective:** In digital electronics, a NAND gate (NOT-AND) is a logic gate which produces an output which is false only if all its inputs are true; thus its output is complement to that of an AND gate. A LOW (0) output results only if all the inputs to the gate are HIGH (1); if any input is LOW (0), a HIGH (1) output results. A NAND gate is made using transistors and junction diodes. By De Morgan's laws, a two-input NAND gate's logic may be expressed as AB=A+B, making a NAND gate equivalent to inverters followed by an OR gate.

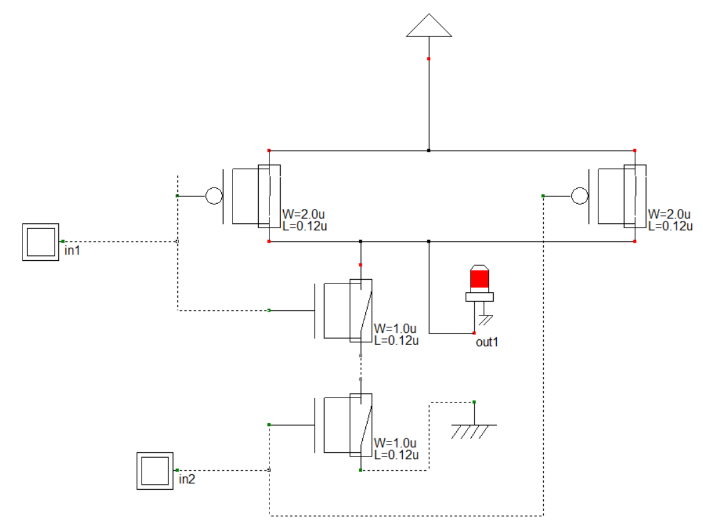
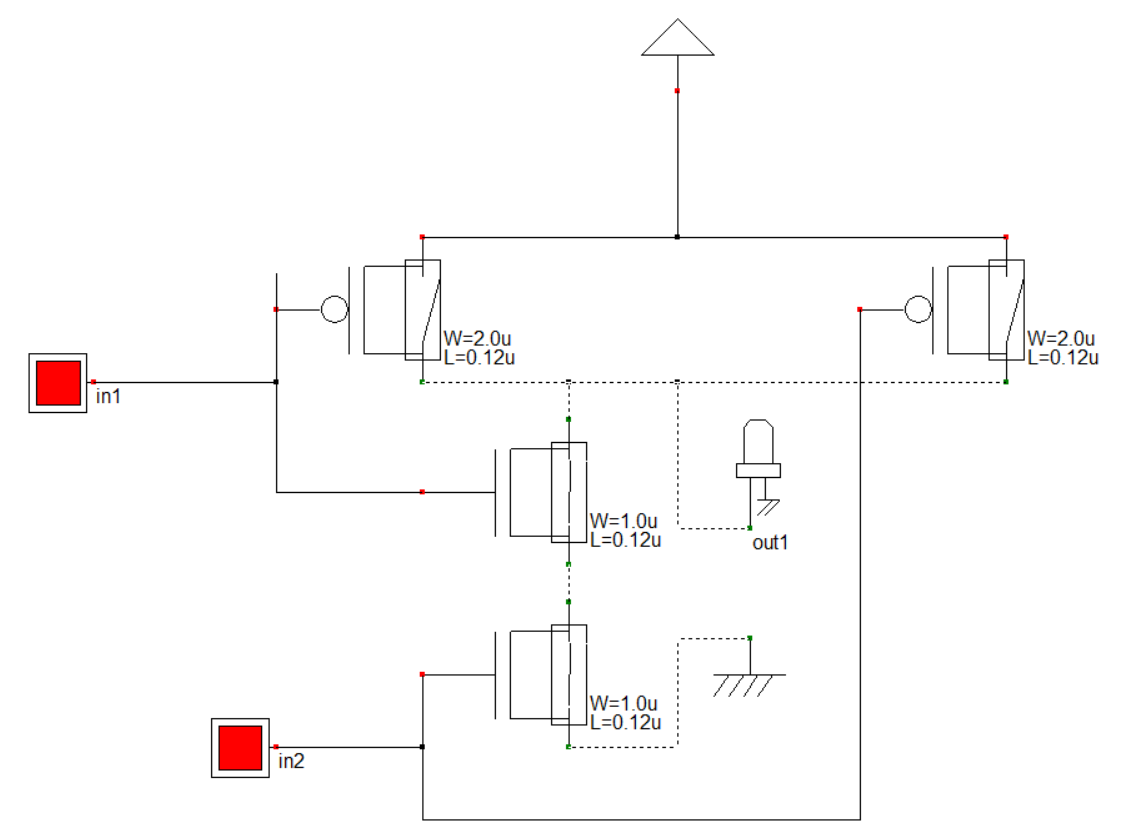
**Theory:**

**Truth table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **Q1** | **Q2** | **Q3** | **Q4** | **Output** |
| 0 | 0 | ON | ON | OFF | OFF | 1 |
| 0 | 1 | ON | OFF | OFF | ON | 1 |
| 1 | 0 | OFF | ON | ON | OFF | 1 |
| 1 | 1 | OFF | OFF | ON | ON | 0 |

**Schematic Diagram:**

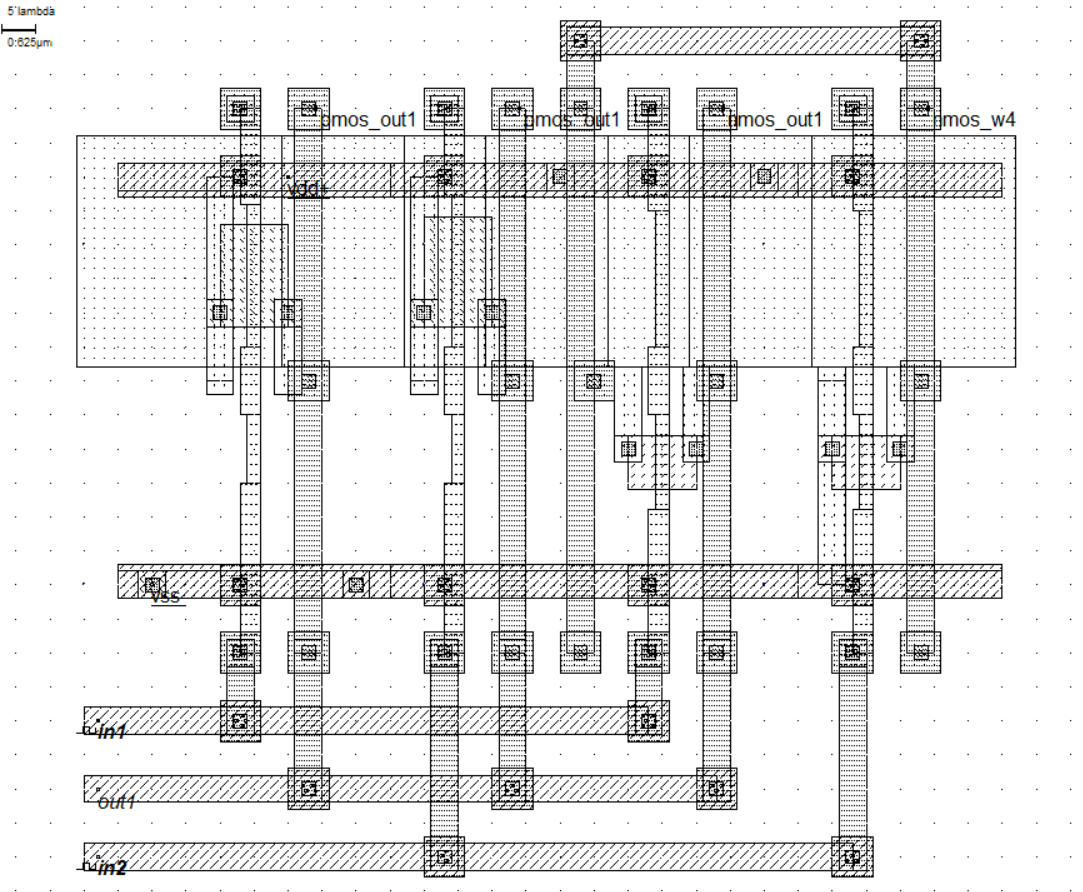
****

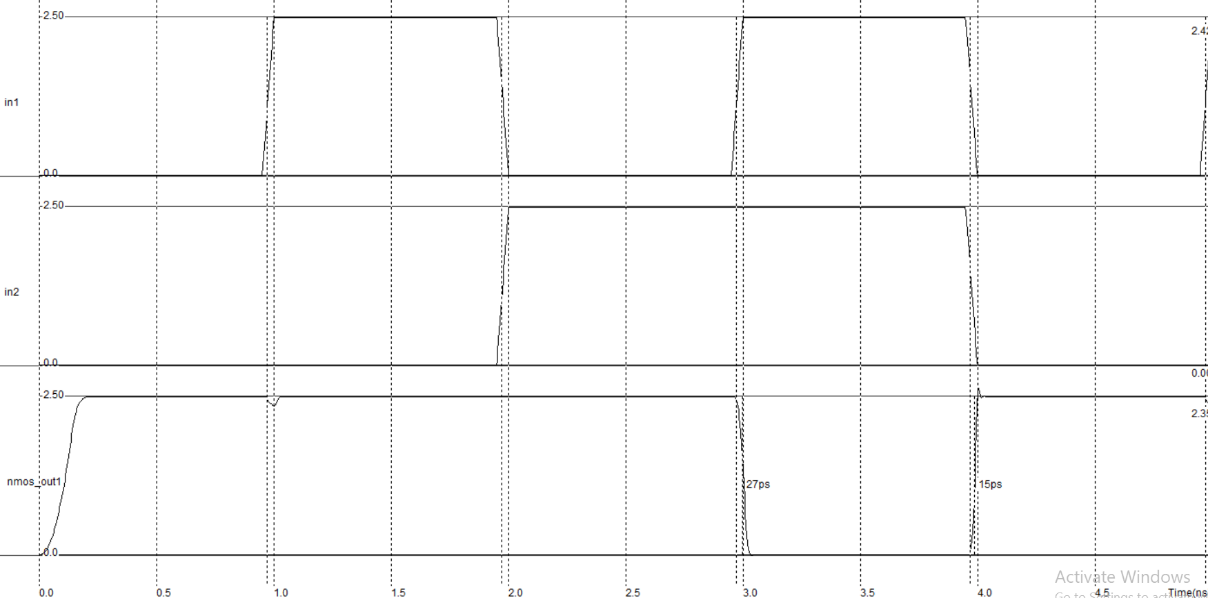
****

**Procedure:**

1. First we open DHCH2 software.
2. 2 NMOS and 2 PMOS gate, 1 VDD, 1 VSS, 1 output, 2 inputs have been taken and connected with wires.
3. Then we check the circuit by voltage on and off to the inputs.
4. Next we make the Verilog file of the circuit and compile the file in Microwind.
5. Finally we simulate the circuit and can see the timing diagrams.

**Result:**

****

****

**Conclusion:** CMOS NAND gate consists of two series NMOS transistors between Y and Ground and two parallel PMOS transistors between Y and VDD. If either input A or B is logic 0, at least one of the NMOS transistors will be OFF, breaking the path from Y to Ground.

**Experiment No:** 2

**Experiment Name:** Implementation of CMOS NOR gate.

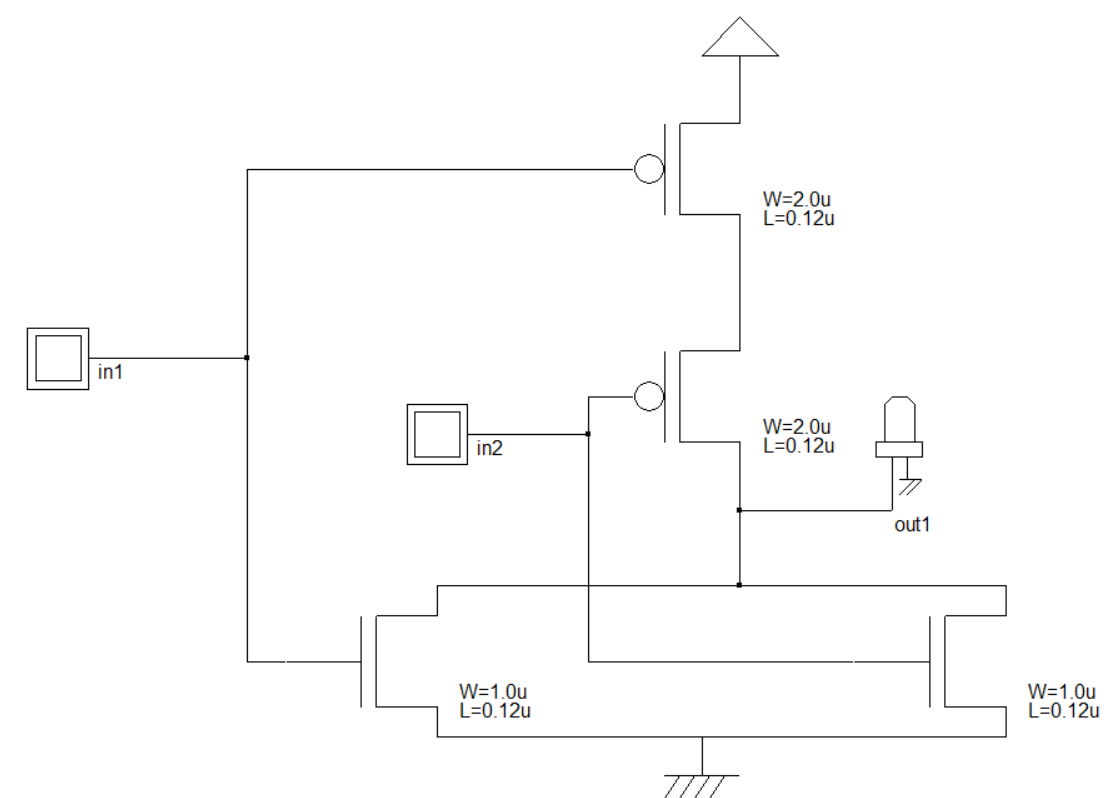
**Objective:** The NOR gate is a digital logic gate that implements logical NOR -it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator. It can also in some senses be seen as the inverse of an AND gate. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. It shares this property with the NAND gate. By contrast, the OR operator is monotonic as it can only change LOW to HIGH but not vice versa.

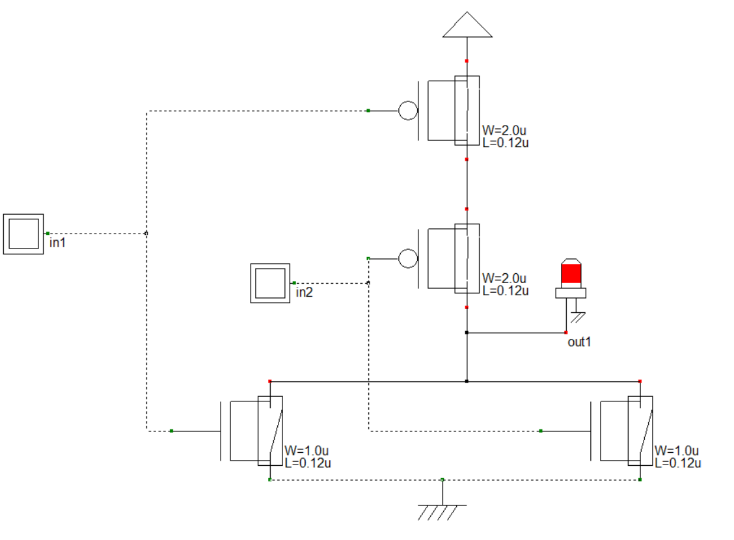
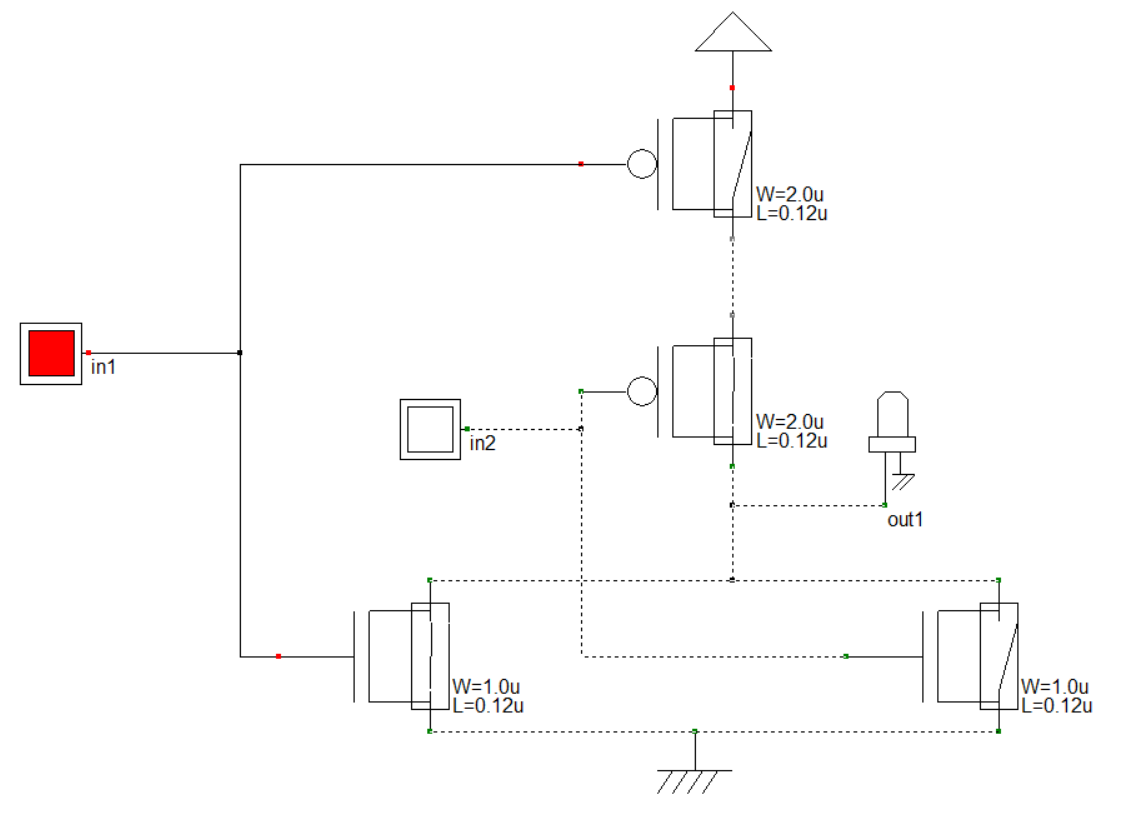
**Theory:**

**Truth table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **Q1** | **Q2** | **Q3** | **Q4** | **Output** |
| 0 | 0 | ON | ON | OFF | OFF | 1 |
| 0 | 1 | ON | OFF | OFF | ON | 1 |
| 1 | 0 | OFF | ON | ON | OFF | 1 |
| 1 | 1 | OFF | OFF | ON | ON | 0 |

**Schematic Diagram:**

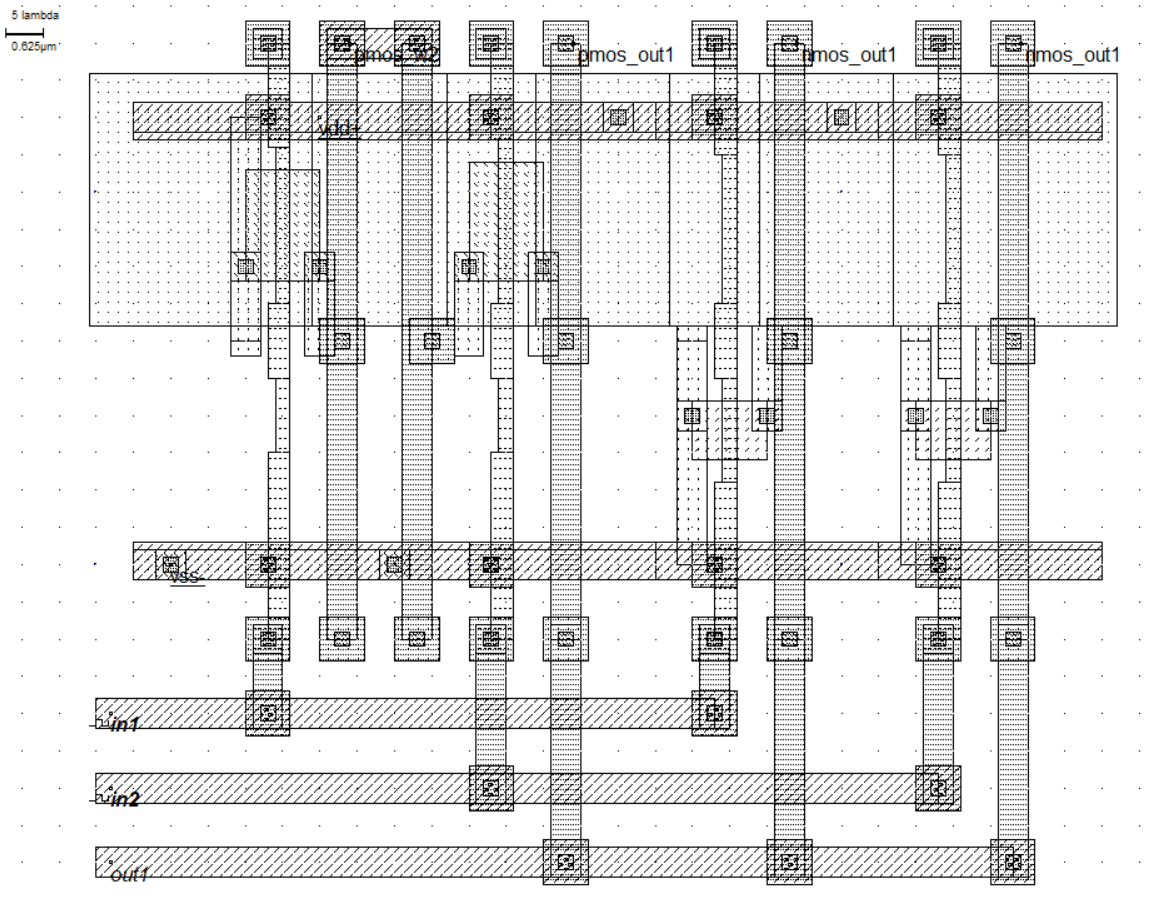


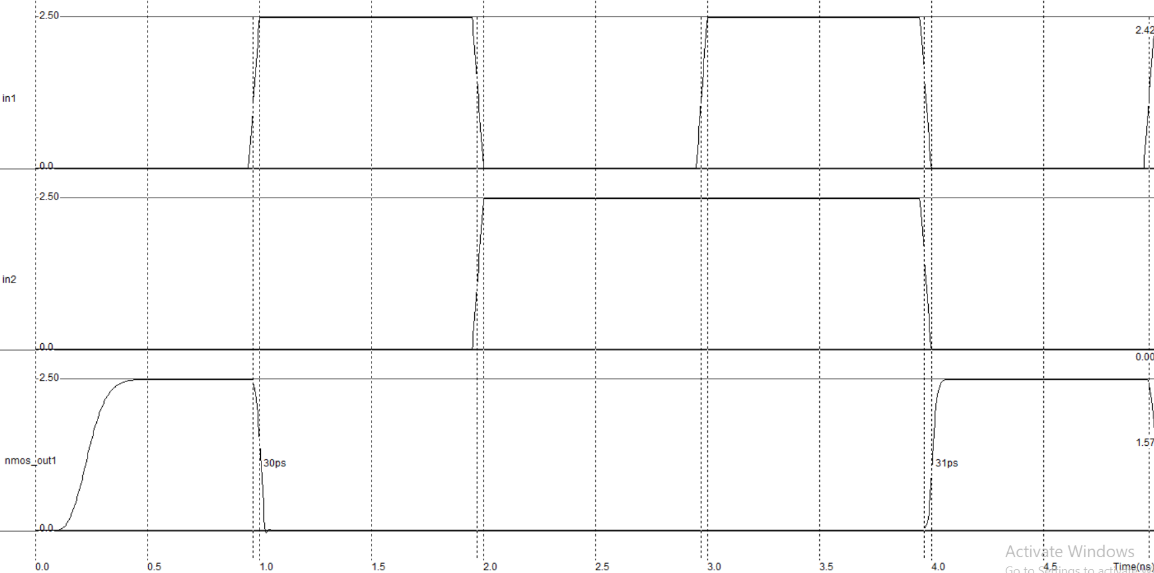


**Procedure:**

1. First we open DHCH2 software.
2. 2 NMOS and 2 PMOS gate, 1 VDD, 1 VSS, 1 output, 2 inputs have been taken and connected with wires.
3. Then we check the circuit by voltage on and off to the inputs.
4. Next we make the Verilog file of the circuit and compile the file in Microwind.
5. Finally we simulate the circuit and can see the timing diagrams.

**Result:**

****

****

**Conclusion:** CMOS NOR gate consists of two series PMOS transistors between Y and Ground and two parallel NMOS transistors between Y and VDD. If either input A or B is logic 1, at least one of the NMOS transistors will be OFF, breaking the path from Y to Ground.

**Experiment No:** 3

**Experiment Name:** Layout design of NAND gate.

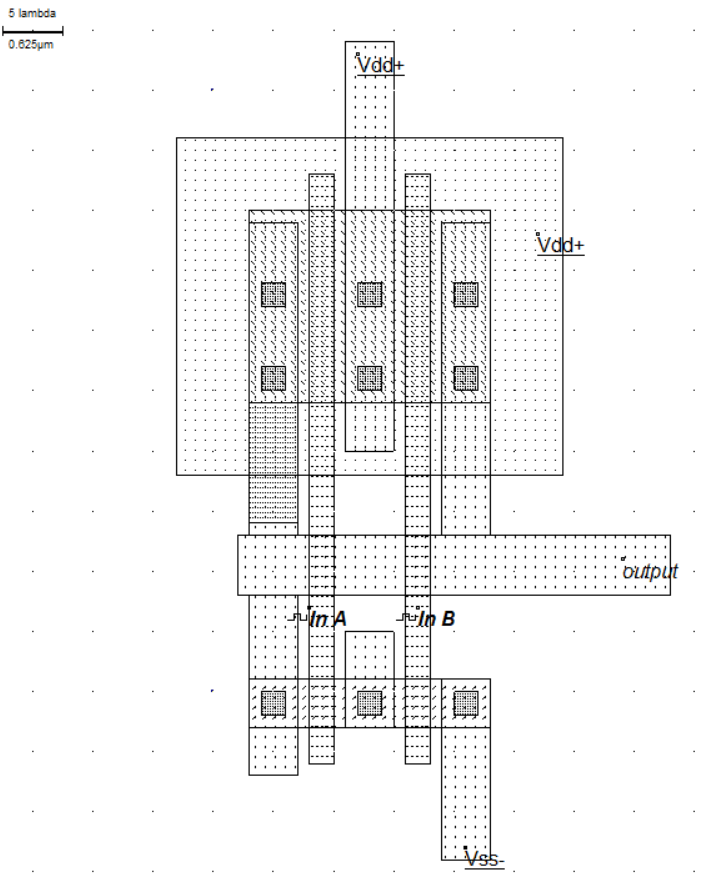
**Objective:** In digital electronics, a NAND gate (NOT-AND) is a logic gate which produces an output which is false only if all its inputs are true; thus its output is complement to that of an AND gate. A LOW (0) output results only if all the inputs to the gate are HIGH (1); if any input is LOW (0), a HIGH (1) output results. A NAND gate is made using transistors and junction diodes. By De Morgan's laws, a two-input NAND gate's logic may be expressed as AB=A+B, making a NAND gate equivalent to inverters followed by an OR gate.

**Theory:**

**Truth table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

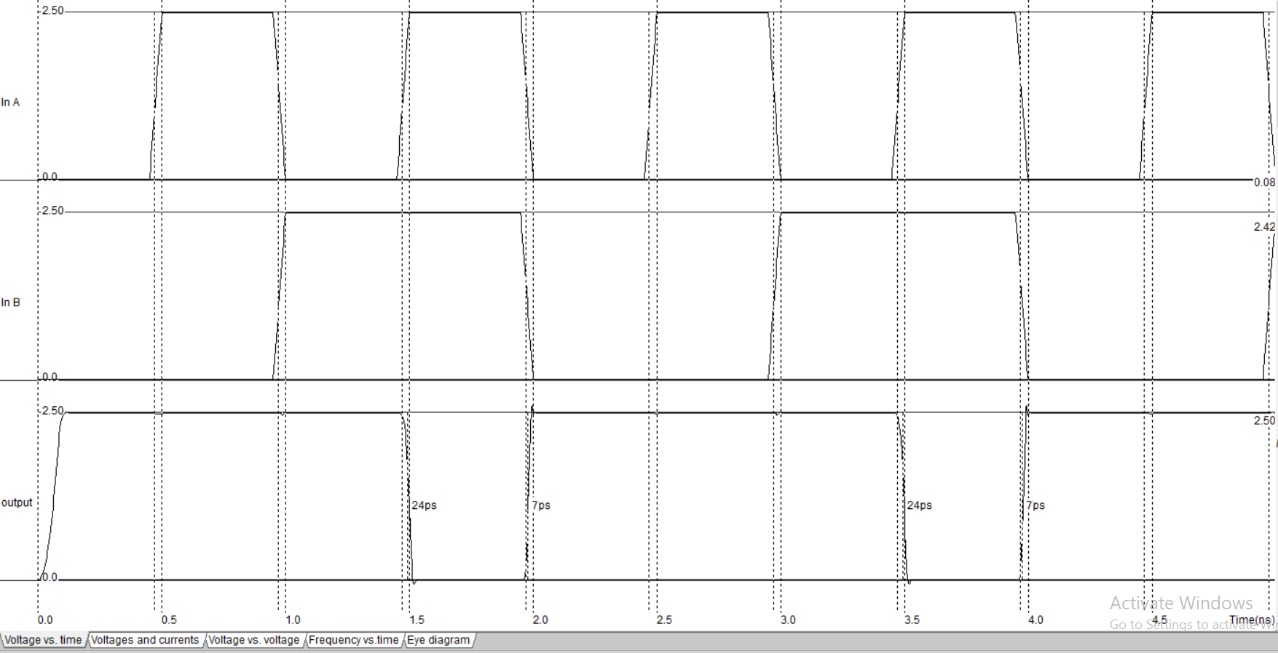
**Schematic Diagram:**



**Procedure:**

1. First we open Microwind2 software.
2. A NMOS of width 4, height 2 and a PMOS of width 16, height 2 in lambda have been taken. They have been connected with Metal 1 and Polysilicon.
3. VDD supply, VSS and 2 input A and B have been added.
4. Finally we simulate the circuit and can see the timing diagrams.

**Result:**



**Conclusion:** NAND gate consists of a NMOS and a PMOS transistors between VSS and VDD. If either input A or B is logic 1, NMOS will be ON making the output ON. And for A and B both being 1, output will be OFF.

**Experiment No:** 4

**Experiment Name:** Layout design of NOR gate.

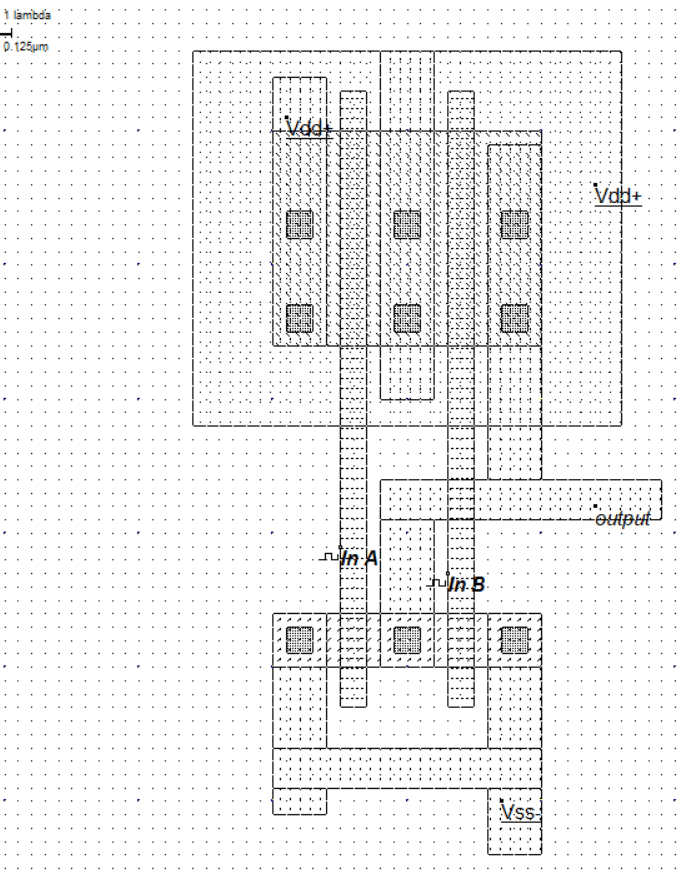
**Objective:** The NOR gate is a digital logic gate that implements logical NOR -it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator. It can also in some senses be seen as the inverse of an AND gate. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. It shares this property with the NAND gate. By contrast, the OR operator is monotonic as it can only change LOW to HIGH but not vice versa.

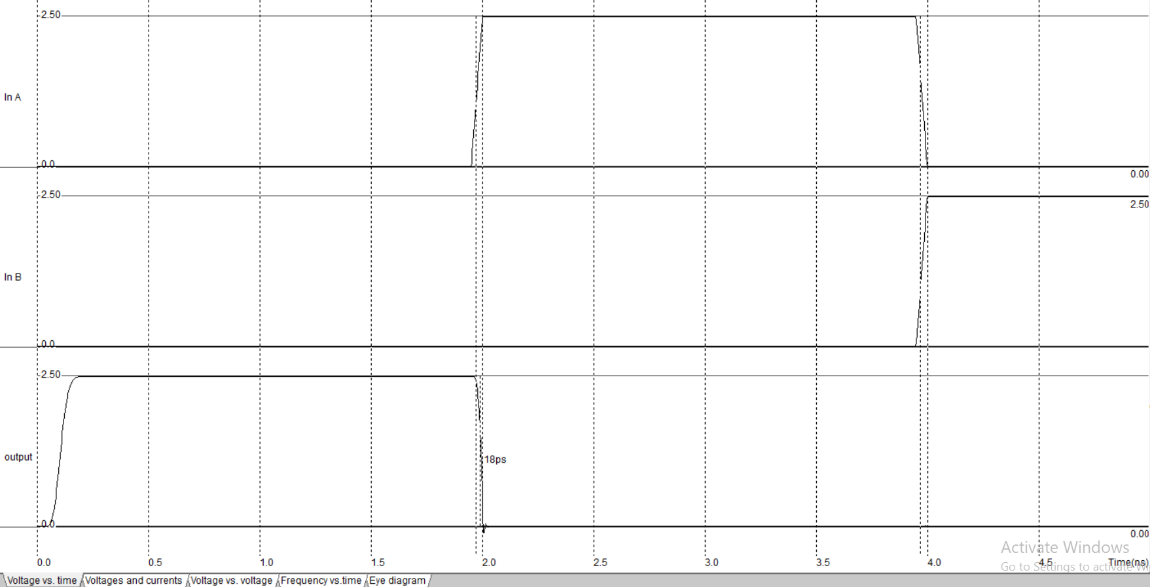
**Theory:**

**Truth table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Schematic Diagram:**





**Conclusion:** NOR gate consists of a NMOS and a PMOS transistors between VSS and VDD. If either input A or B is logic 1, PMOS will be OFF making the output OFF. And for A and B both being 0, output will be ON.

**Experiment No:** 4

**Experiment Name:** Verification of XOR gate using DHCH2 and Microwind2.

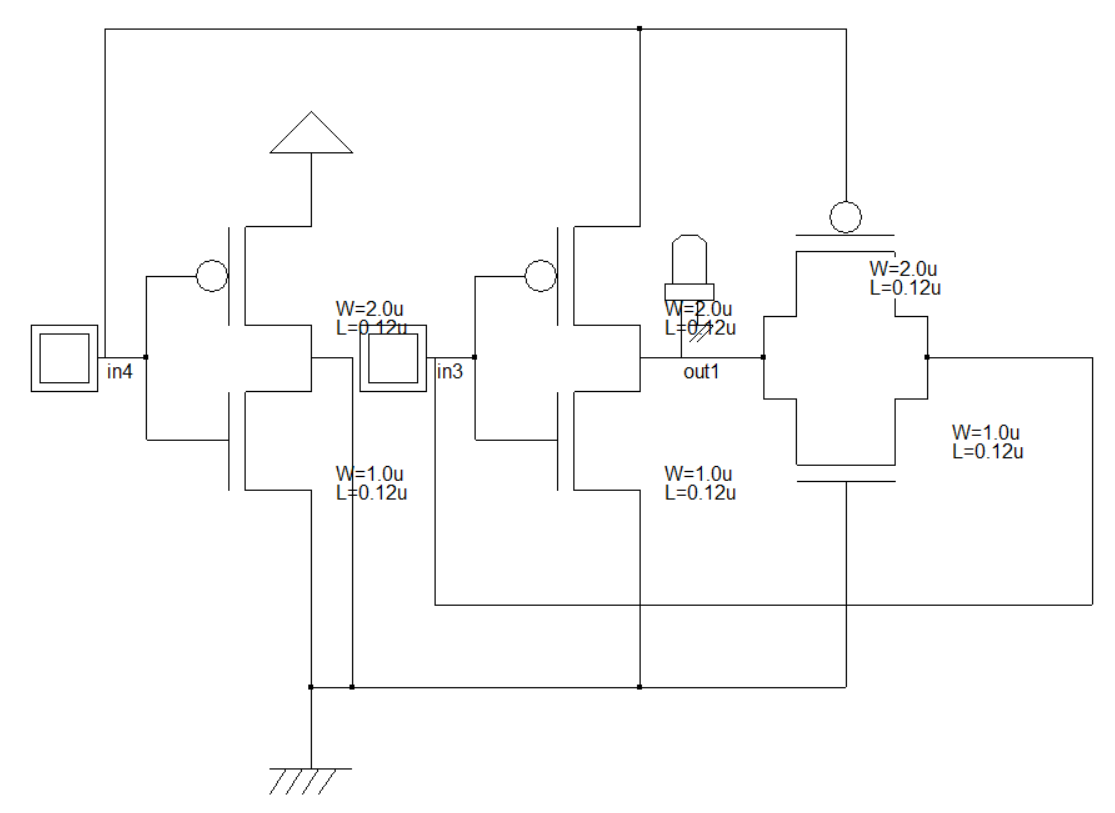
**Objective:** XOR gate (sometimes EOR, or EXOR and pronounced as Exclusive OR) is a digital logic gate that gives a true (1 or HIGH) output when the number of true inputs is odd. An XOR gate implements an exclusive or from mathematical logic; that is, a true output results if one, and only one, of the inputs to the gate is true. If both inputs are false (0/LOW) or both are true, a false output results.

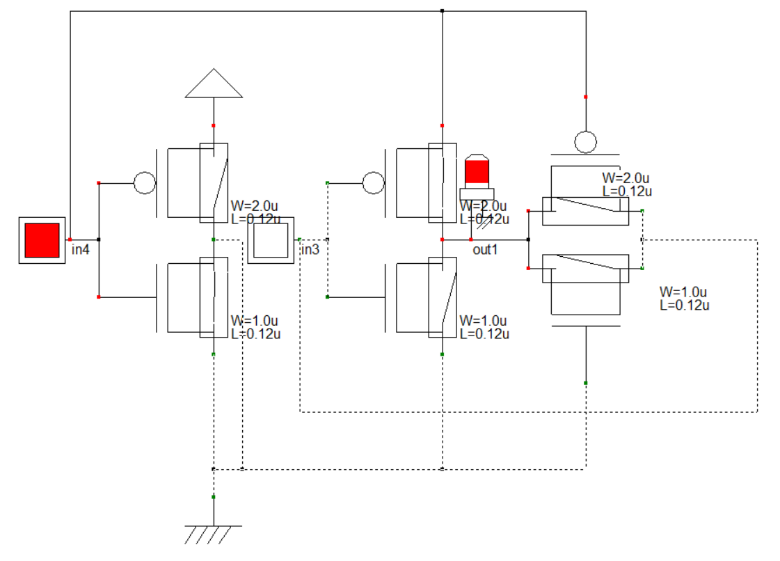
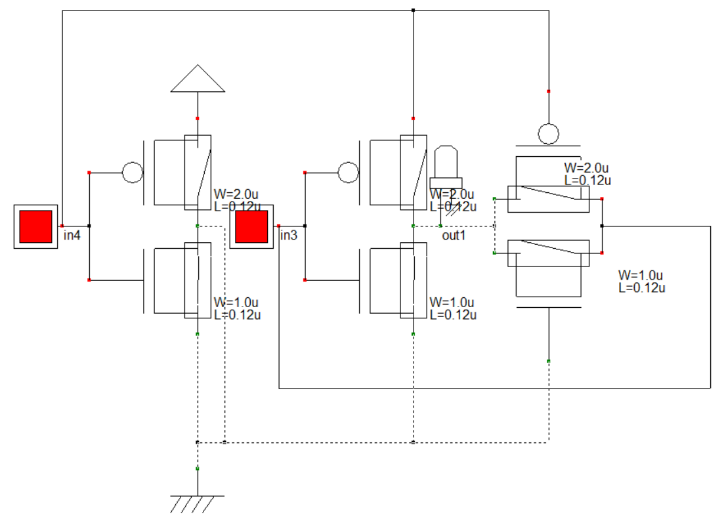
**Theory:**

**Truth table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Schematic Diagram:**

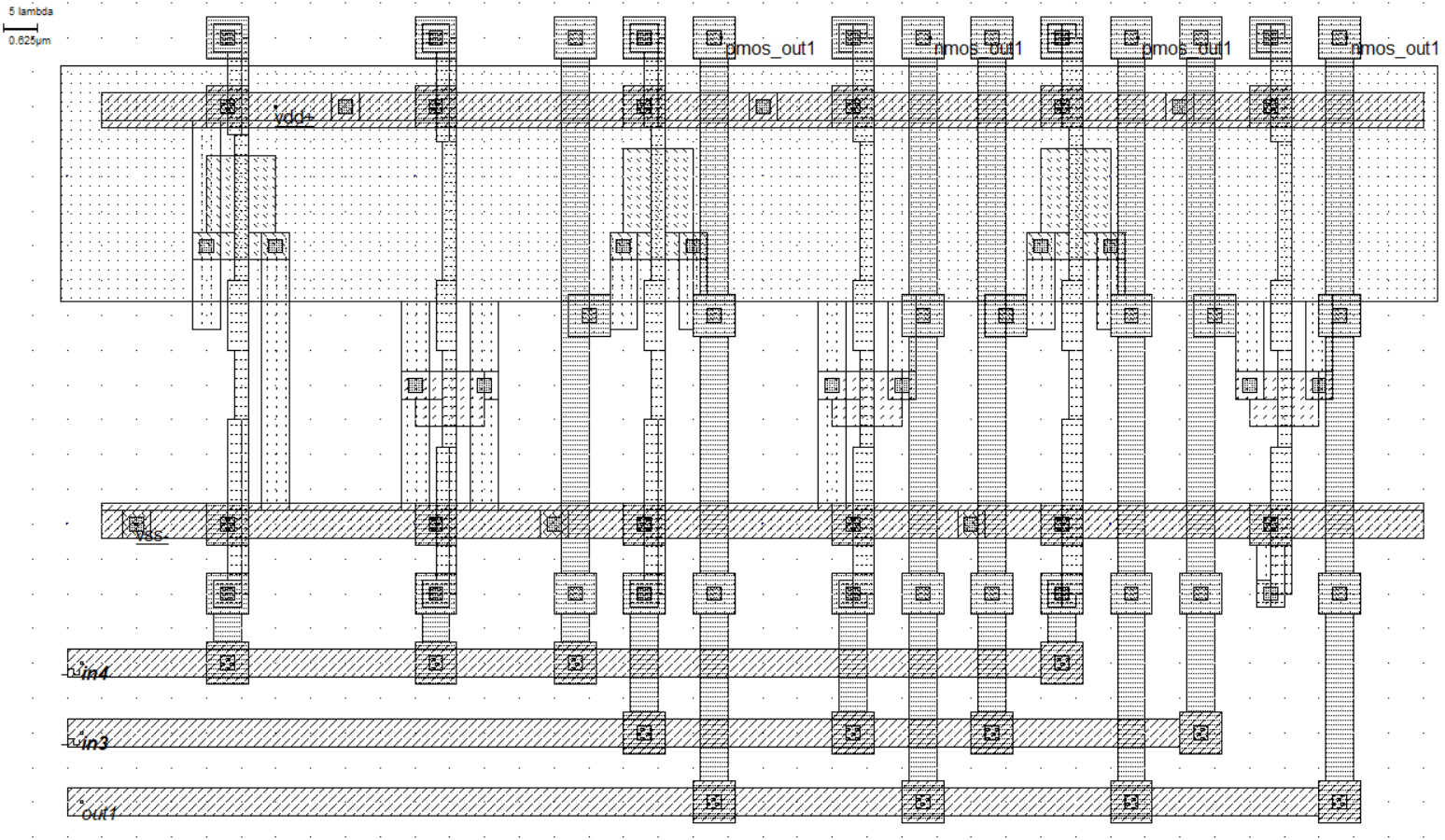


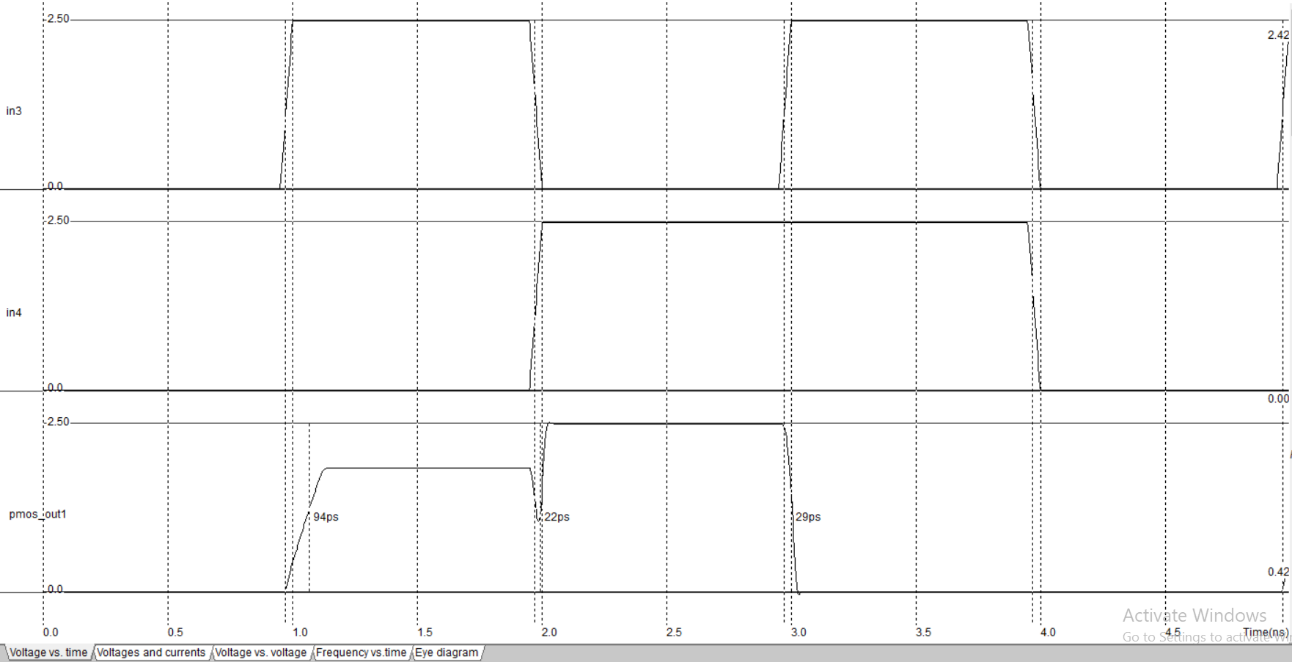


**Procedure:**

1. First we open DHCH2 software.
2. 3 NMOS and 3 PMOS gate, 1 VDD, 1 VSS, 1 output, 2 inputs have been taken and connected with wires.
3. Then we check the circuit by voltage on and off to the inputs.
4. Next we make the Verilog file of the circuit and compile the file in Microwind2.
5. Finally we simulate the circuit and can see the timing diagrams.

**Result:**

****

****

**Conclusion:** XOR represents the inequality function, i.e., the output is true if the inputs are not alike otherwise the output is false. A way to remember XOR is "must have one or the other but not both".

**Experiment No:** 5

**Experiment Name:** Implementation of Buffer using 2 inverters.

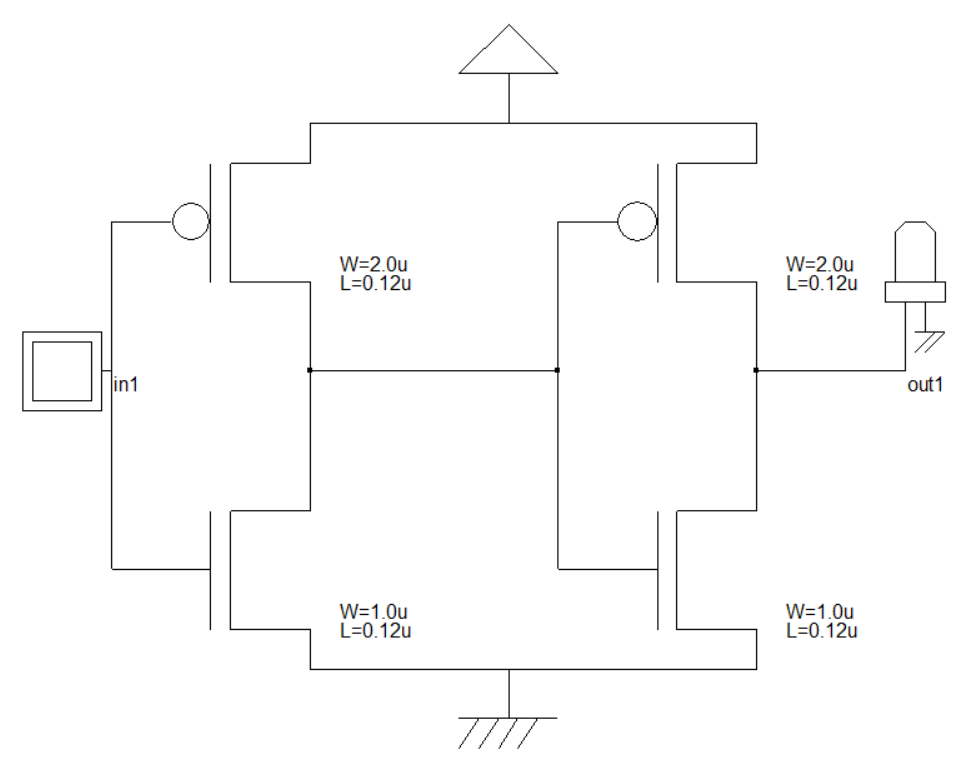
**Objective:** A buffer, is a basic logic gate that passes its input, unchanged, to its output. Its behavior is the opposite of a NOT gate. The main purpose of a buffer is to regenerate the input, usually using a strong high and a strong low. A buffer has one input and one output; its output always equals its input.

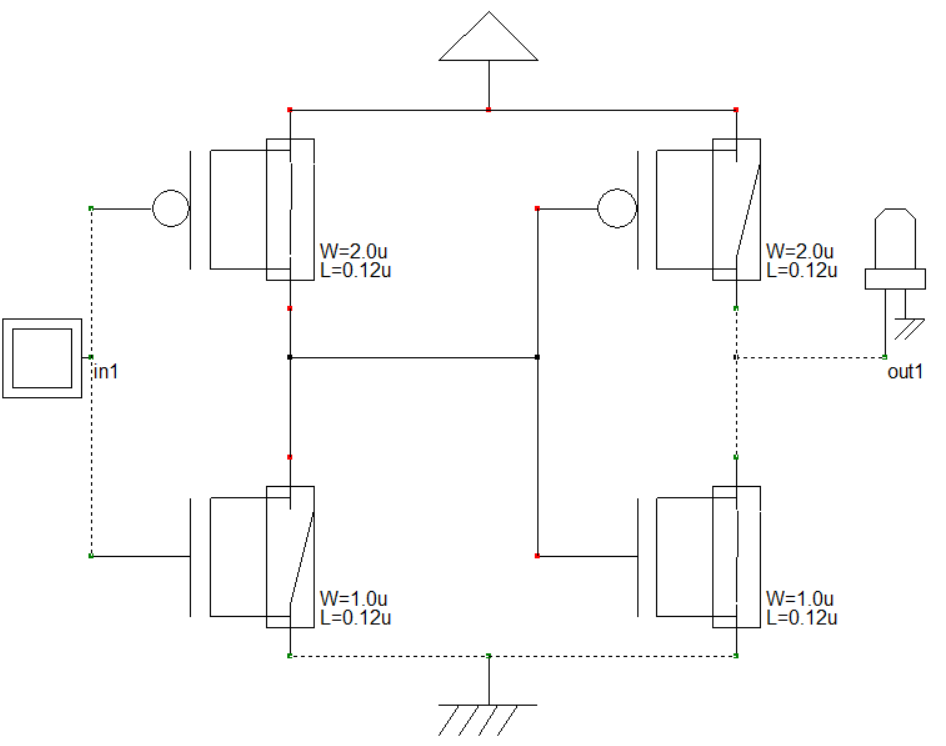
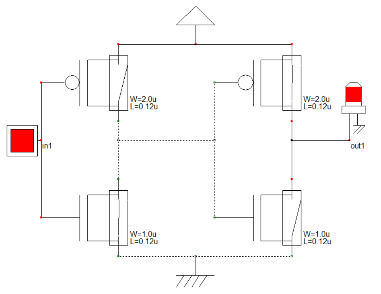
**Theory:**

**Truth table:**

|  |  |
| --- | --- |
| **Vin** | **Vout** |
| 0 | 0 |
| 0 | 1 |

**Schematic Diagram:**

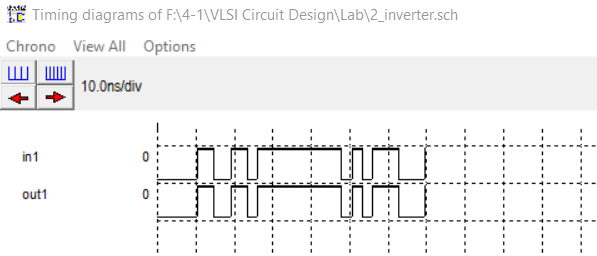


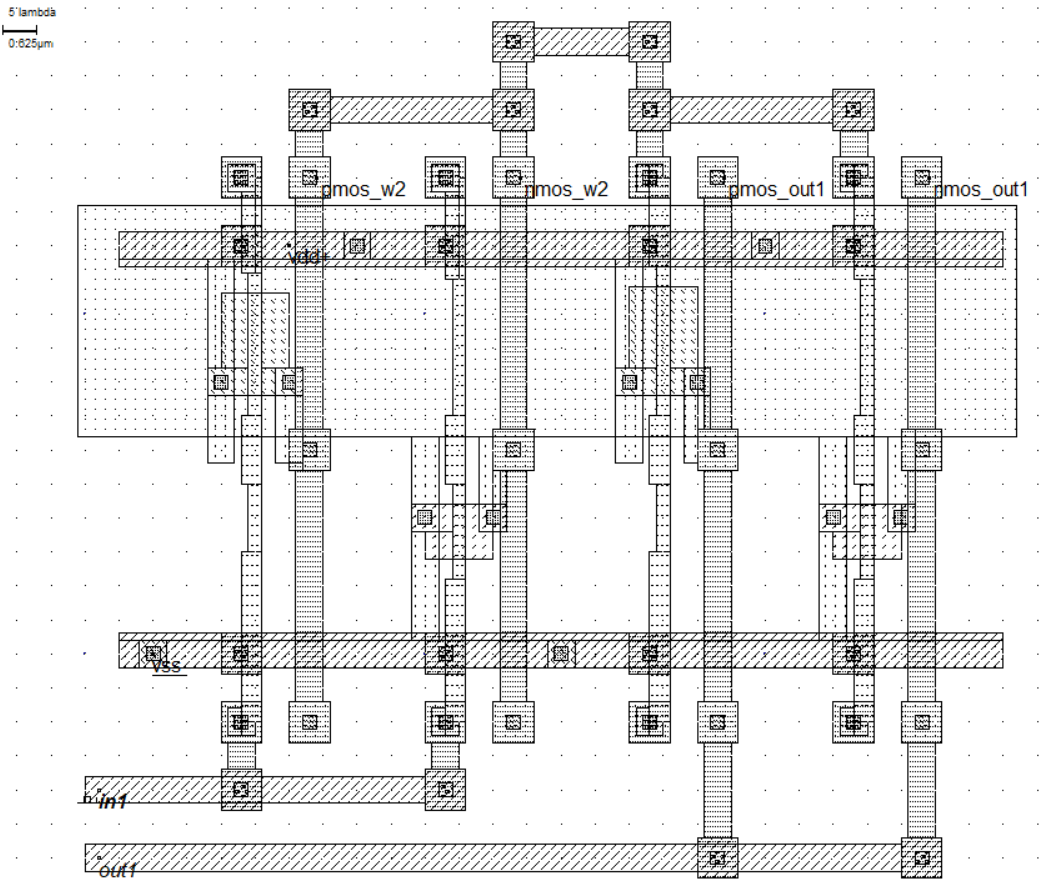


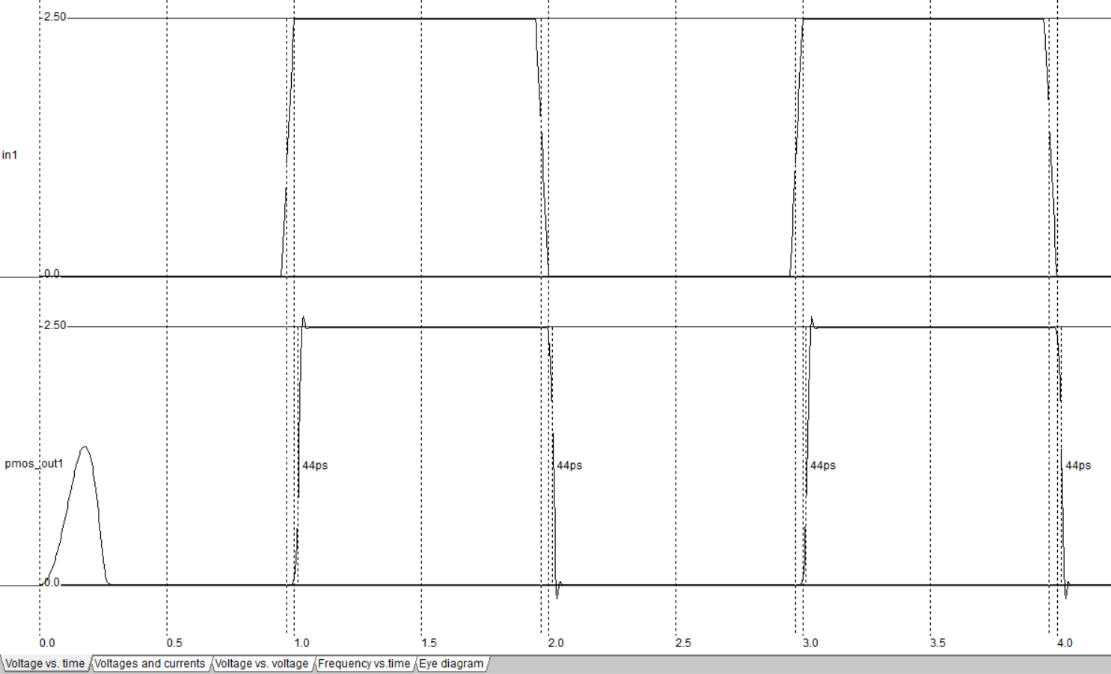
**Procedure:**

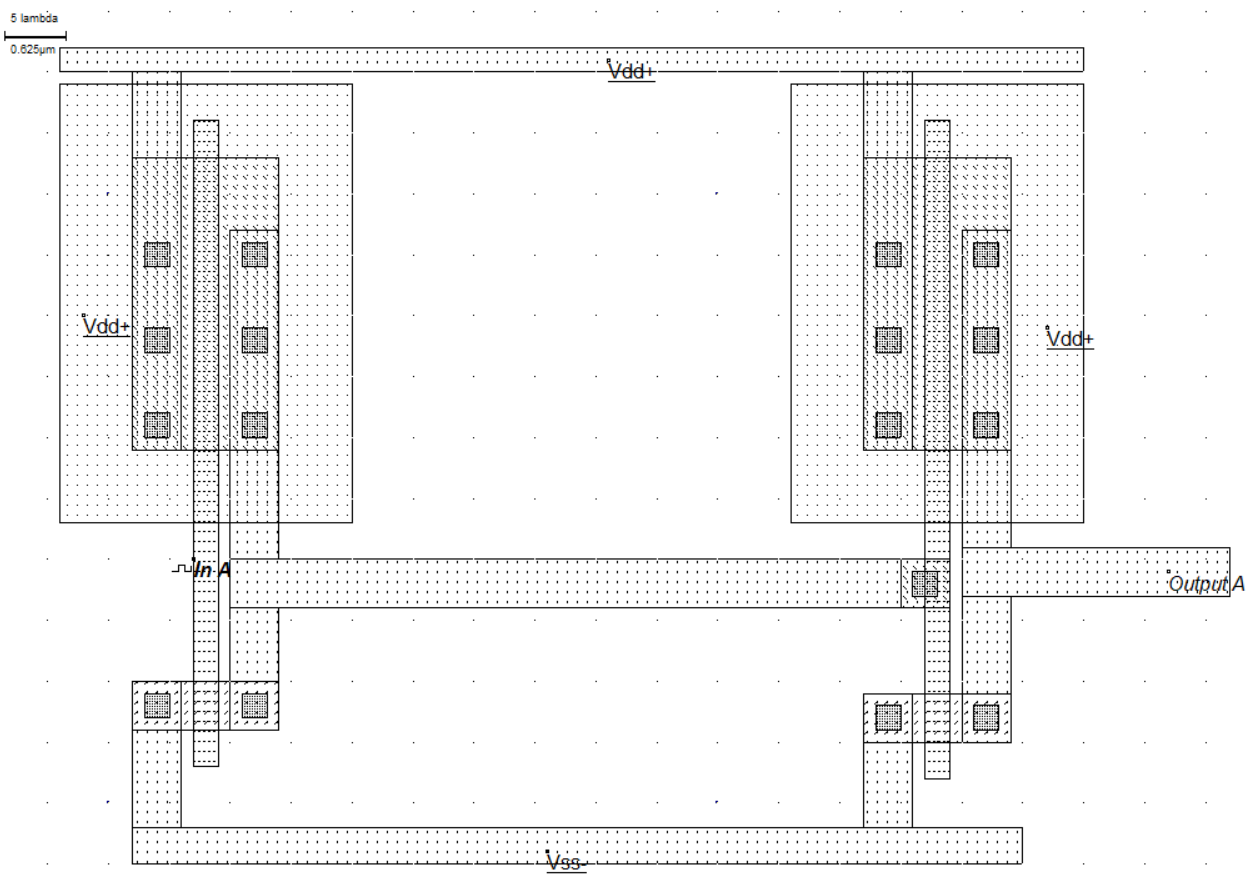
1. First we open DHCH2 software.
2. 2 NMOS and 2 PMOS gate, 1 VDD, 1 VSS, 1 output, 1 input have been taken and connected with wires.
3. Then we check the circuit by voltage on and off to the inputs.
4. Next we make the Verilog file of the circuit and compile the file in Microwind2.
5. Finally we simulate the circuit and can see the timing diagrams.
6. Again, we open Microwind2 software.
7. A NMOS of width 4, height 2 and a PMOS of width 24, height 2 in lambda have been taken. They have been connected with Metal 1 and Polysilicon.
8. VDD supply, VSS and an input have been added.
9. Finally we simulate the circuit and can see the timing diagrams.

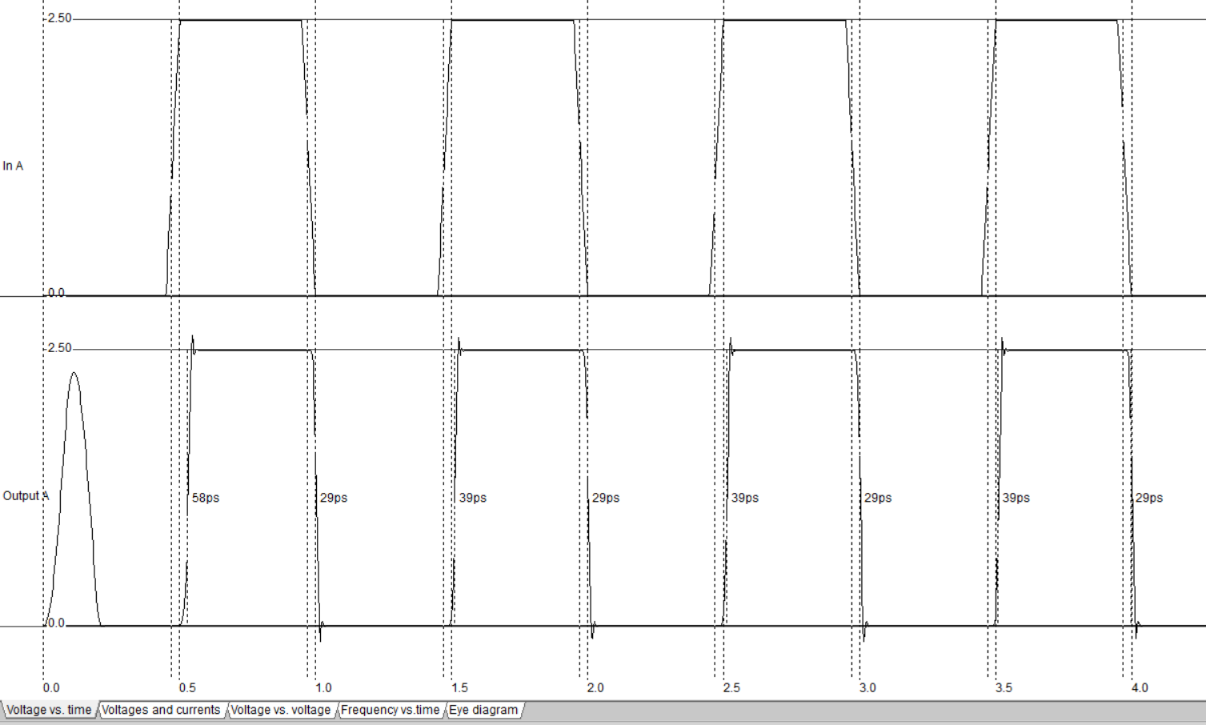
**Result:**











**Conclusion:** CMOS acts like an inverter so that for input 0 the transistor will be ON and for input 1 the transistor will be OFF. Connecting two CMOS together will result the same as the input.

**Experiment No:** 6

**Experiment Name:** Observation and transfer characteristics of CMOS inverter.

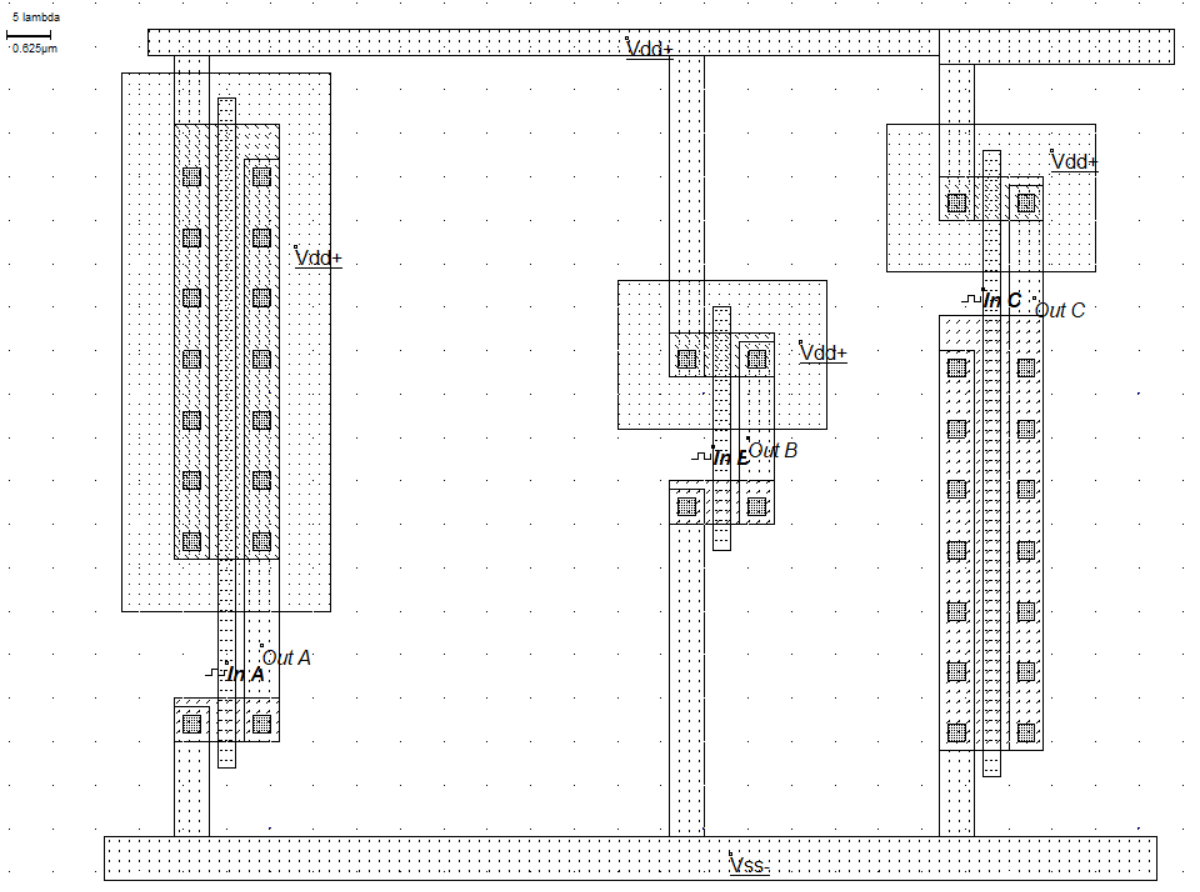
**Objective:** A buffer, is a basic logic gate that passes its input, unchanged, to its output. Its behavior is the opposite of a NOT gate. The main purpose of a buffer is to regenerate the input, usually using a strong high and a strong low. A buffer has one input and one output; its output always equals its input.

**Theory:**

**Truth table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Wpmos** | **Lpmos** | **Wnmos** | **Lnmos** | **βn/βp** |
| 1st | 50 | 1 | 5 | 1 | 0.1 |
| 2nd | 5 | 1 | 5 | 1 | 1 |
| 3rd | 5 | 1 | 50 | 1 | 10 |

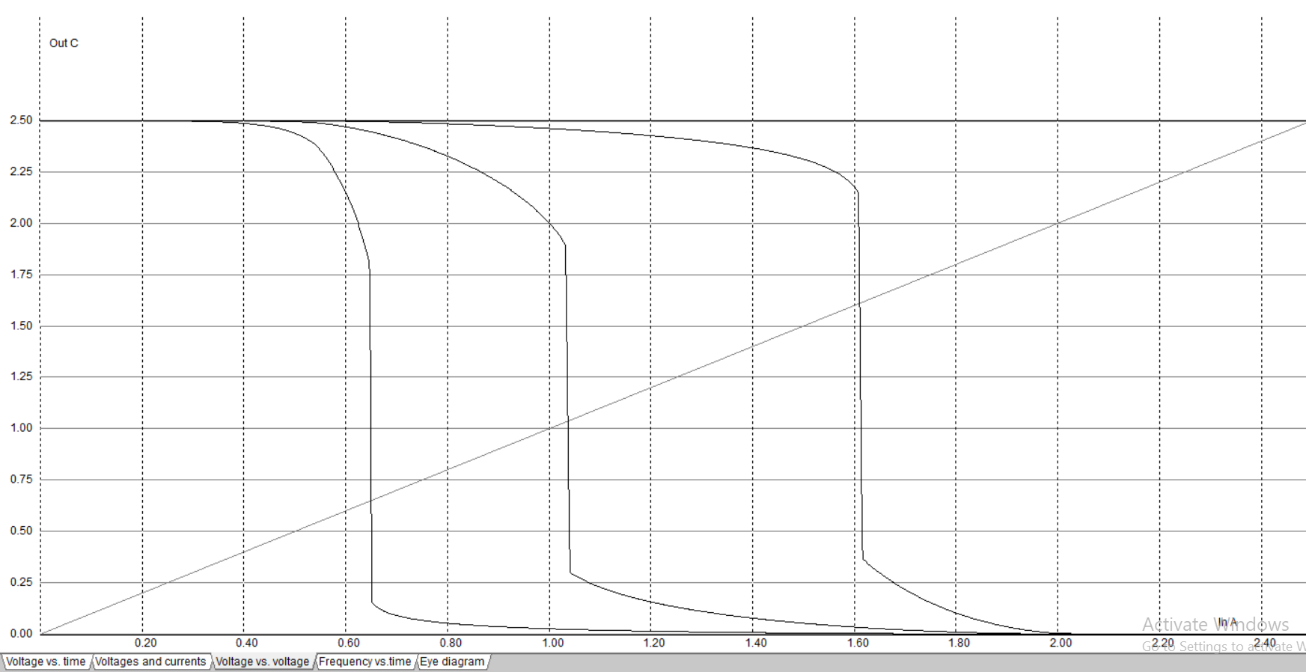
**Schematic Diagram:**



**Procedure:**

1. First we open Microwind2.
2. A PMOS of width 50, length 1 and a NMOS of width 5, length 1 in lambda have been taken. They have been connected with Metal 1 and Polysilicon.
3. Again a PMOS of width 5, length 1 and a NMOS of width 5, length 1 in lambda have been taken and connected with Metal 1 and Polysilicon.
4. Again a PMOS of width 5, length 1 and a NMOS of width 50, length 1 in lambda have been taken and connected with Metal 1 and Polysilicon.
5. VDD supply, VSS and an input have been added.
6. Finally we simulate the circuit and can see the timing diagrams.

**Result:**



**Conclusion:** Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Significant power is only drawn while the transistors in the CMOS device are switching between on and off states.

**Experiment No:** 7

**Experiment Name:** Verification of nMOS and pMOS DC-Characteristics.

**Objective:** Use circuit simulator of Microwind2 to do a DC simulation of the IDS current vs VDS voltage and calculation of the threshold voltage.

**Theory:** The nMOS transistor IDS =0 versus VDS voltage equations are as follows:

Cut-off mode: IDS =0 when VGS <0

Triode/Linear region: IDS=kn{(VGS-VTn)VDS-1/2 VDS2} when VDS<VGS-VTn ….(1)

In Level 1 SPICE model, IDS=UO.ε0εSiO2/TOX W/L{(VGS-VTn)VDS-1/2 VDS2}

Saturation

egion: IDS=1/2 kn{(VGS-VTn)2(1+λVDS)} when VDS>VGS-VTn ….(2)

In Level 1 SPICE model, IDS=1/2 UO.ε0εSiO2/TOX W/L(VGS-VTn)2

When the channel modulation effect is neglected the drain current equation (2) can be simplified as IDS=1/2 kn{(VGS-VTn)2} when VDS>VGS-VTn ….(3)

For pMOS:

Triode/Linear region: IDS=kn{(VGS-VTn)VDS-1/2 VDS2} when VDS>VGS-VTn ….(1)

Saturation region: IDS=1/2 kn{(VGS-VTn)2(1+λVDS)} when VDS<VGS-VTn ….(2)

When the channel modulation effect is neglected the drain current equation (2) can be simplified as IDS=1/2 kn{(VGS-VTn)2} when VDS>VGS-VTn ….(3)

**Procedure:**

a) Level 1 MOS model equations to calculate DC values for the drain current IDS vs drain-source voltage VDS (paper and pencil)

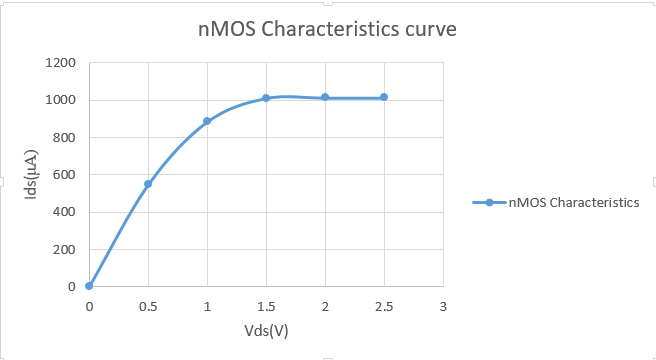
1. First calculate with a gate-source voltage VGS = +2.0V for nMOS and VGS = -2.0V for pMOS, for each drain-source voltage VDS: 0.5V, 1.0V, 1.5V, 2.0V, 2.5V the difference VDS - (VGS - VTn) to determine in which region the transistor work. Mark in Table equation number (1) or (2) that should be chosen to calculate the drain current IDS for the corresponding point (IDS ; VDS ). If we consider the .25μm CMOS process then Vth = 0.45 V, μ0 = μn = 0.06, γ=0.4 for nMOS and Vth = -0.45 V, μ0 = μn =- 0.06, γ=-0.4 for pMOS.
2. Calculate for the voltages VDS: 0.5V, 1.0V, 1.5V, 2.0V, 2.5 V the drain current IDS for each point ( IDS ;VDS ) after having determined a region the transistor works in (linear or saturated) Fill in corresponding value pairs (IDS ; VDS ) in Table below for given drain-source DC voltages.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VDS(V) | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
| VDS-(VGS-VTn) | -1.05 | -0.55 | -0.05 | 0.45 | 0.95 |
| (1) | (1) | (1) | (2) | (2) |
| Manual calculation, IDS (μA) | 547.56 | 884.52 | 1010.88 | 1011.933 | 1011.933 |

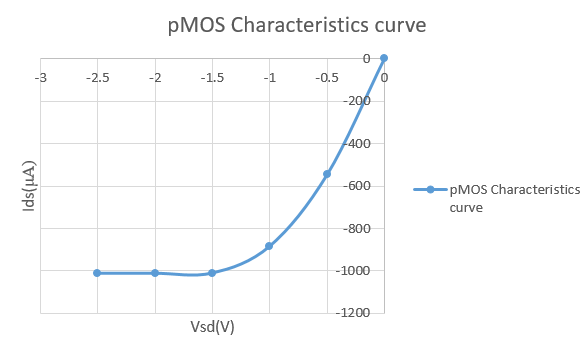
For pMOS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VDS(V) | -0.5 | -1.0 | -1.5 | -2.0 | -2.5 |
| VDS-(VGS-VTn) | 1.05 | 0.55 | 0.05 | -0.45 | -0.95 |
| (1) | (1) | (1) | (2) | (2) |
| Manual calculation, IDS (μA) | -547.56 | -884.52 | -1010.88 | -1011.933 | -1011.933 |

1. Now plot the characteristic curve using excel based on data values of table.

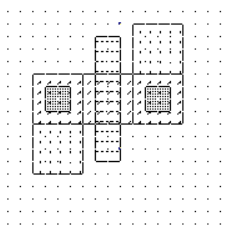


For pMOS:

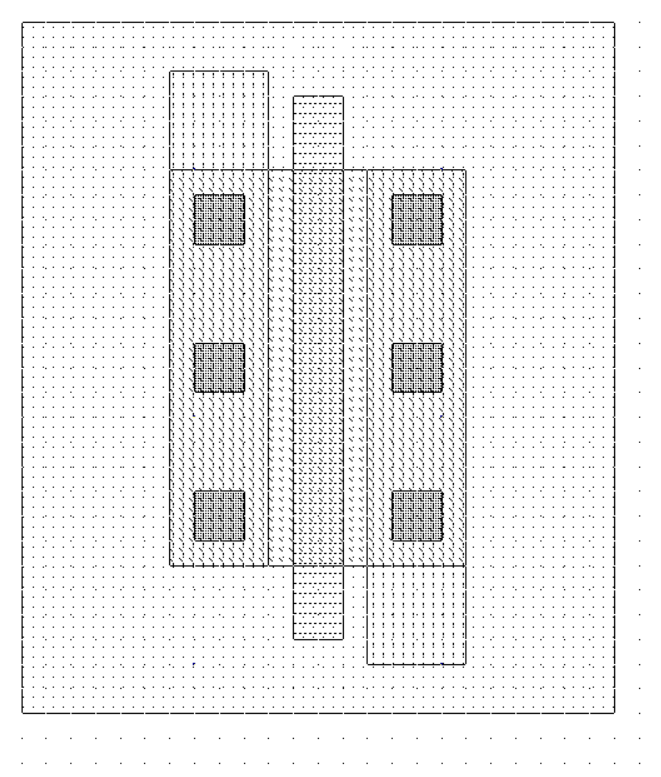


b) Use of “Simulate>MOS characteristic” to generate the DC characteristic IDSn vs VDSn for the NMOS transistor in microwind.

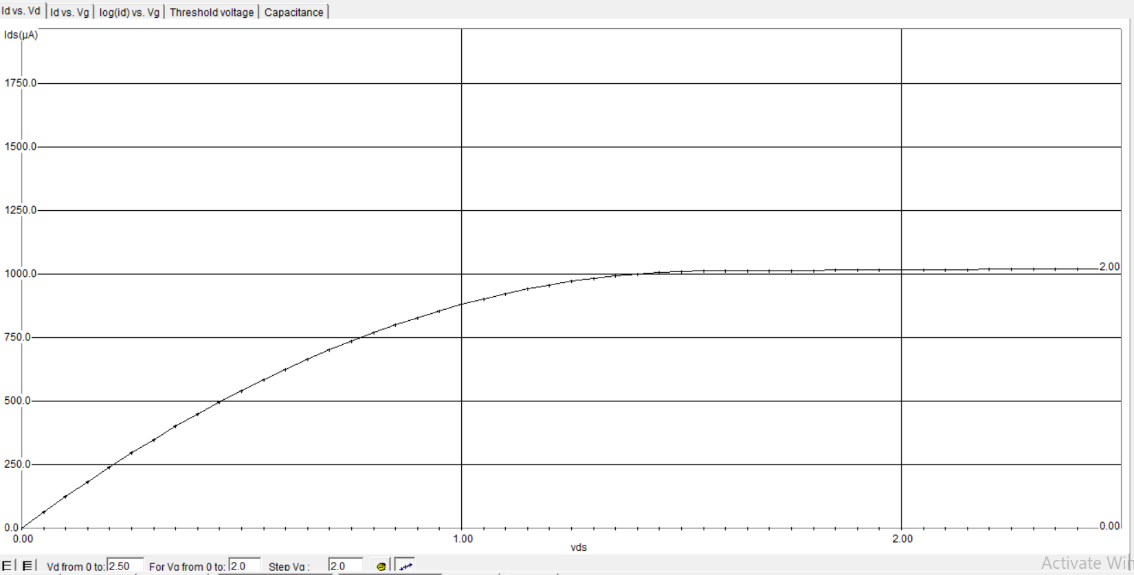
1. Open the microwind window and Check that the selected foundry is 0.25 μm CMOS process from file→select foundry and choose cmos025.rul. Use Level1 MOS transistor model.
2. Now generate the nMOS transistor of width and length ratio is 4γ x 2γ from palate. The nMOS generation process is shown.



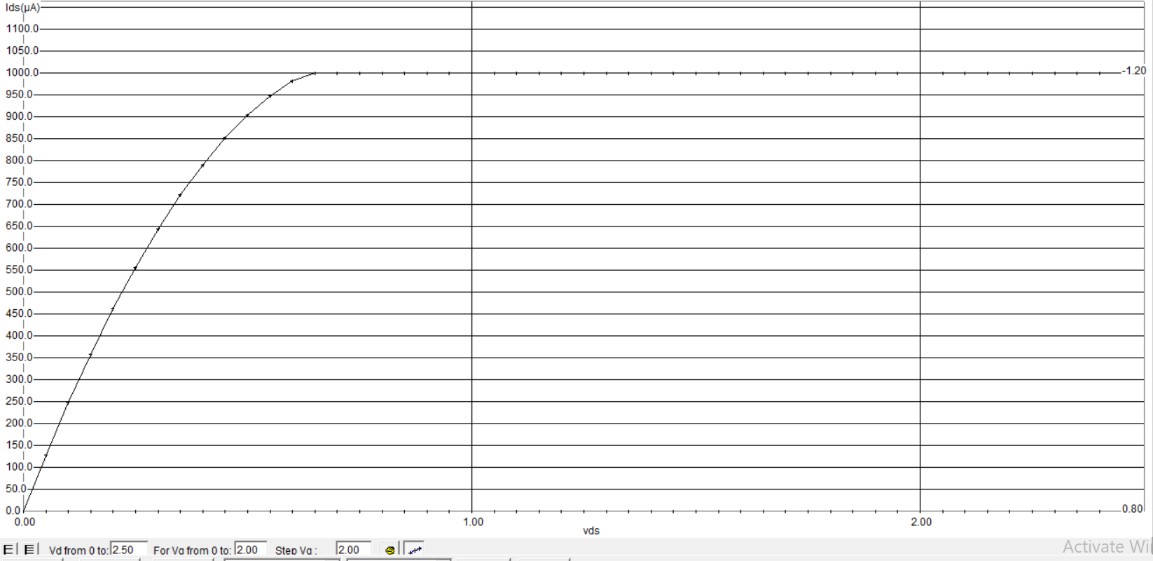
For pMOS:



1. Now click Simulate→MOS characteristics and then point to the NMOS transistor In the upper left corner of the window you can now choose functions to plot. Choose to show for the nMOS-transistor the current IDS vs VDS for VGS = +2.0V.
2. On the bottom row to the left in the plot window, fill in the to-value of Vdd, the to-value of Vg and the step-value of Vg so that you can read the text: “Vd from 0 to 2.50 For Vg from 0 to: 2.00 Step Vg: 2.00”. Then you should push the button Draw to draw the curve.



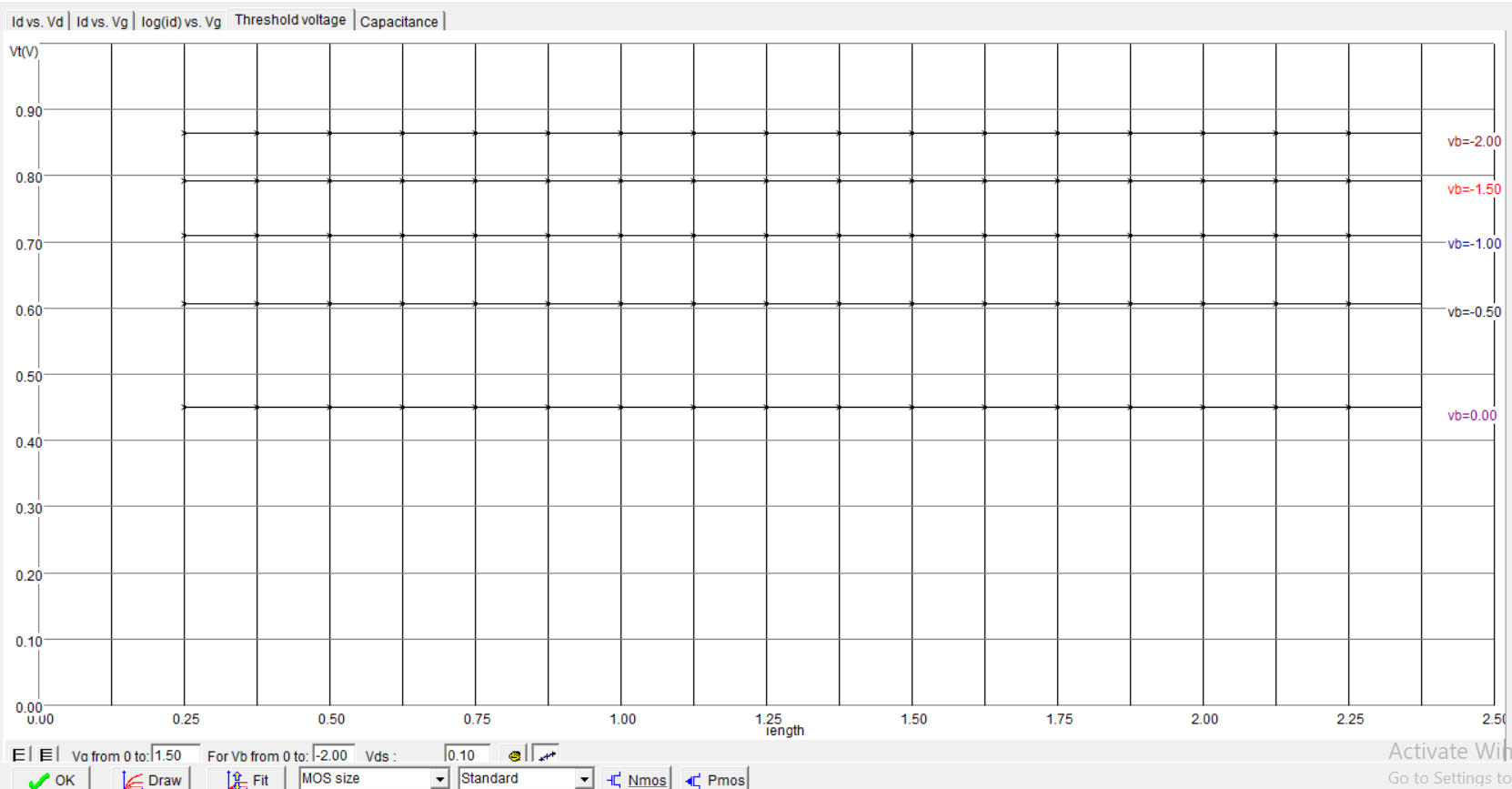
For pMOS:



c) Calculation of the threshold voltage factor (γ)

1. Variation of threshold voltage dependence of the bulk-to-source voltage VSB, called the Body effect. In the upper experiment we have considered that the threshold voltage factor gamma γ=0.

From the graph, we have found that UTO (Threshold voltage under the condition of gamma factor is 0)= 0.45V. This can be shown in the threshold voltage tab in the IDS vs VDS simulation window when the voltage of substrate voltage bias Vsb=0 shown.



**Result:** From both the above calculation we can say that both the manually calculated result and simulated result are close.

**Conclusion:** In this experiment, we have seen DC characteristic of MOS transistor. We can see current IDS vs VDS and threshold voltage in the IDS vs VDS simulation.