# Experiment Name: Design and Implementation of NAND gate using DSCH.

## Objectives:

* To implement NAND gate
* To simulate our design in three different ways:
  1. Circuit design and Logic verification using DSCH2.
  2. Circuit design simulation and Logic verification using DSCH2 and Micro wind.
  3. Layout design simulation using Micro wind.
* To observe the deviation in results with default layout (automatic layout generation procedure) and our manual layout.

**Apparatus:**

**Dsch2/3 software**

**Micro wind software**

## Discription:

**The NAND Gate:**

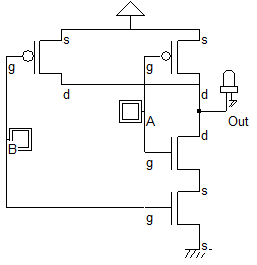
The truth-table and logic symbol of the NAND gate with 2 inputs are shown below. In DSCH, select the NAND symbol in the palette, add two buttons and one lamp as shown above. Add interconnects if necessary to link the button and lamps to the cell pins. Verify the logic behavior of the cell.

**The truth table and symbol of the NAND gate**

|  |  |  |
| --- | --- | --- |
| input1 | input2 | **Output** |
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |



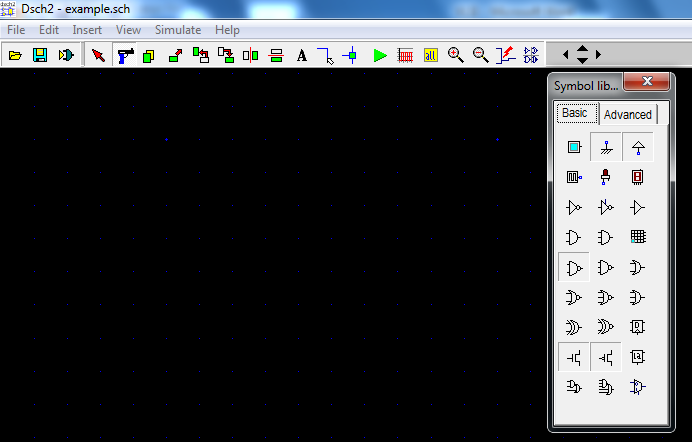
In CMOS design, the NAND gate consists of two nMOS in series connected to two pMOS in parallel. The schematic diagram of the NAND cell is reported below. The nMOS in series tie the output to the ground for one single combination A=1, B=1. For the three other combinations, the nMOS path is cut, but a least one pMOS ties the output to the supply VDD. Notice that both nMOS and pMOS devices are used in their best regime: the nMOS devices pass “0”,the pMOS pass “1”.



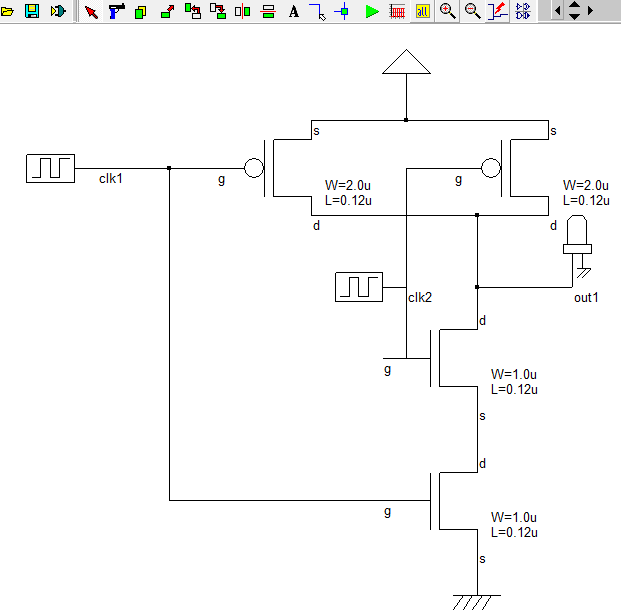
**Fig.: Schematic diagram of NAND gate**.

**Working Procedure:**

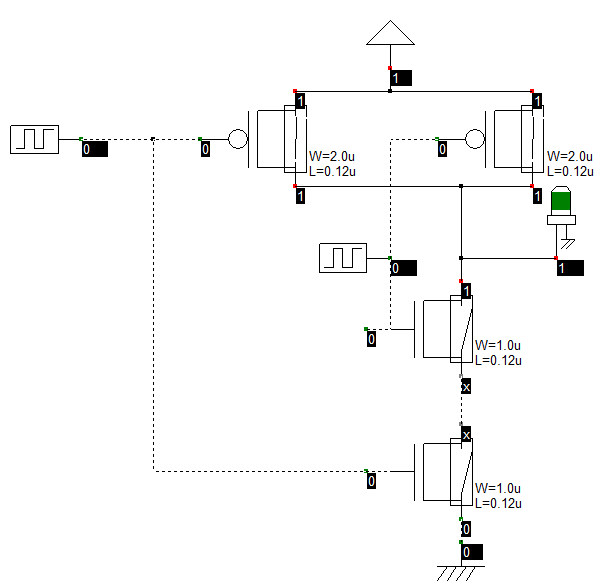
* Open the schematic editor called DSCH2/3. Click on the transistor symbol in the symbol library on the right.



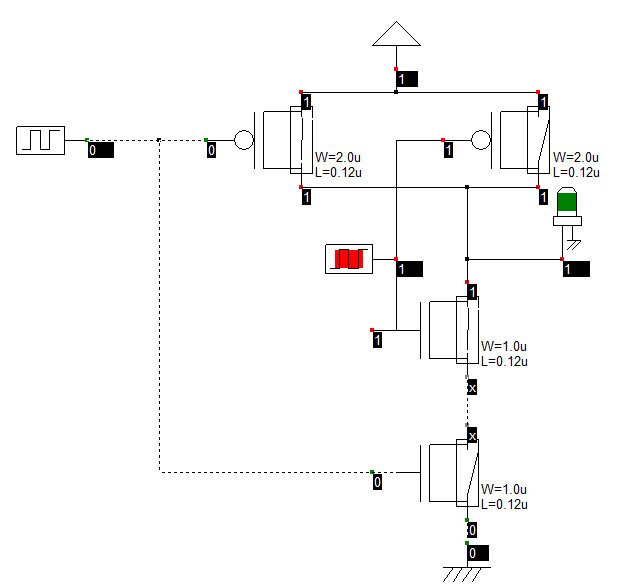
* Instantiate nMOS or pMOS transistor (drag and drop) from the symbol library, place them in the editor window and connect them according to schematic diagram shown below. Connect Vdd, Gnd from the symbol library. Also connect (In1 and In2) and Output (Out). Button or Clock symbol can be used as input and LED symbol can be used as output shown below:



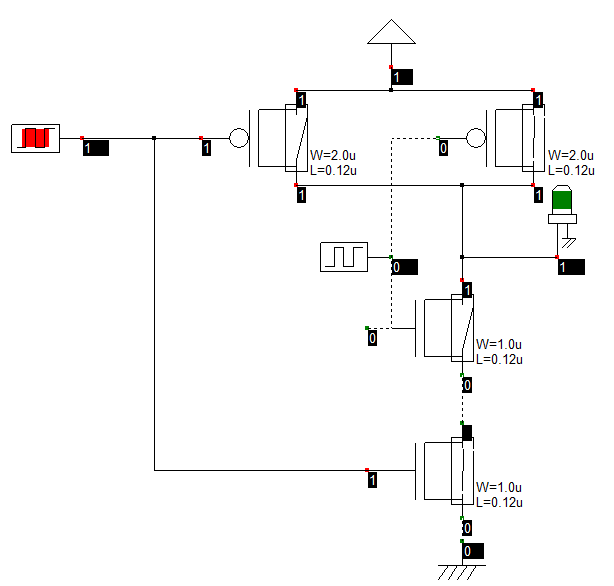
* At this point, the NAND gate is ready to simulate. Before going to simulation, just check the floating line exists in the diagram. For this purpose, just click “check floating line” in the simulation menu. The next step is to simulate and check the logic functionality. Click on Simulate🡪Start simulation. This brings up a simulation Control window with Simulation speed settings. Click on the input buttons to set them to 1 or 0. Red color in a switch indicates logic 1. The output should be changed according to the input changes.
* **Input : 0 0**

******

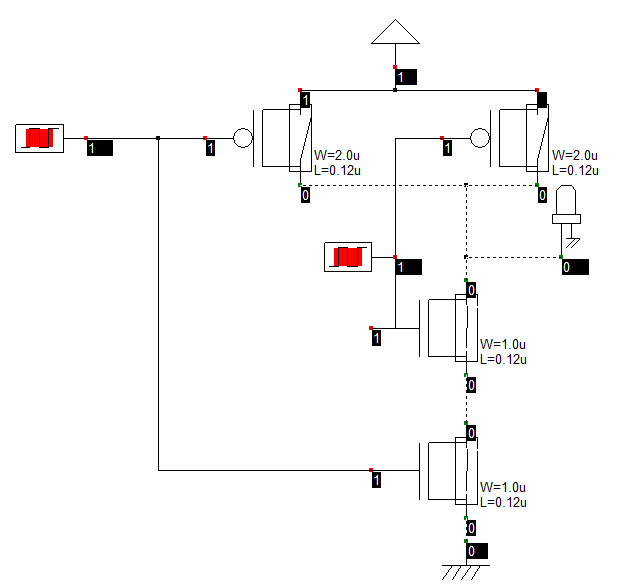
* **Input : 0 1**



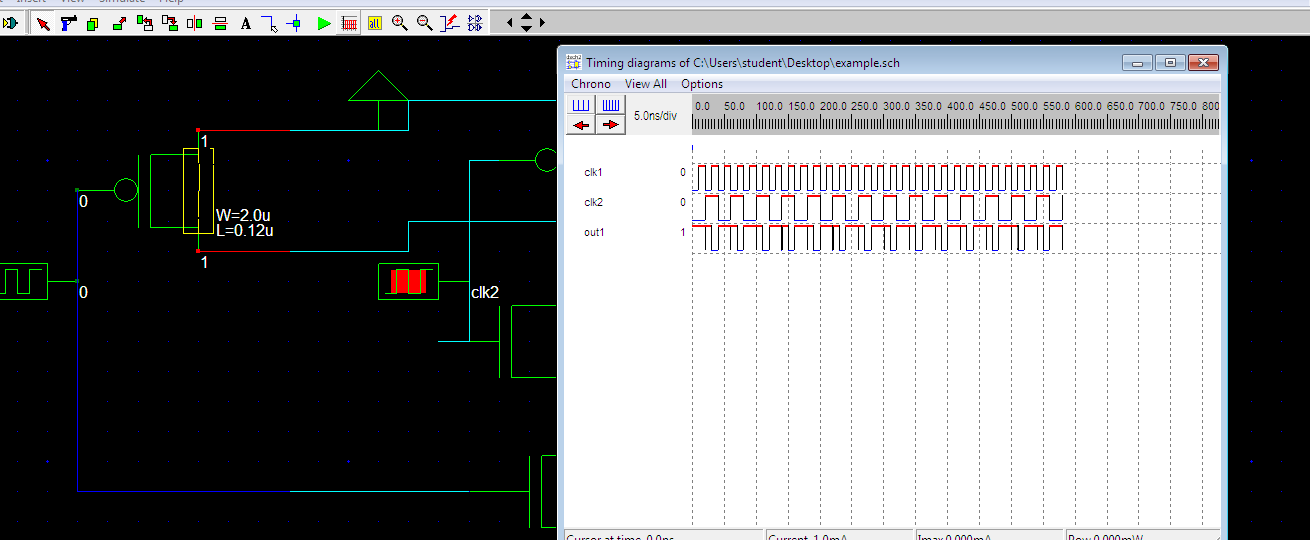
* **Input : 1 0**



* **Input : 1 1**



The above figures show the functionality of the NAND gate according to the truth table. The simulation output can be observed after the application of the inputs as above. Click on the (last icon) timing diagram icon in the icon menu to see the timing diagram of inputs and outputs.



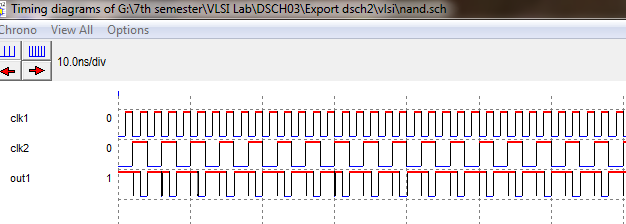
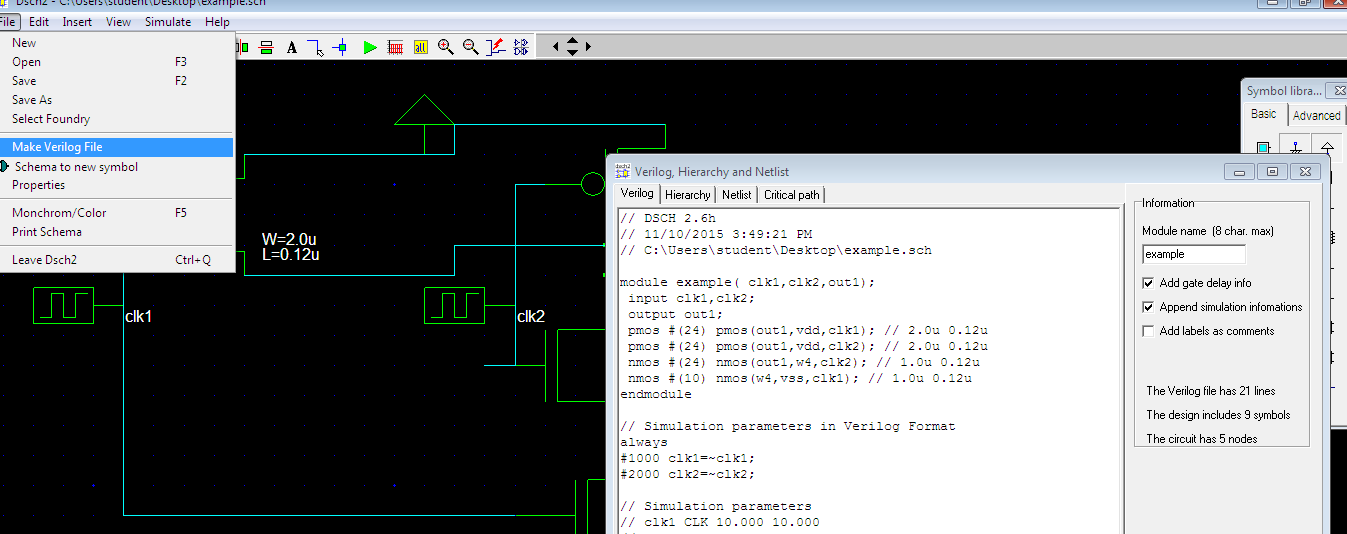
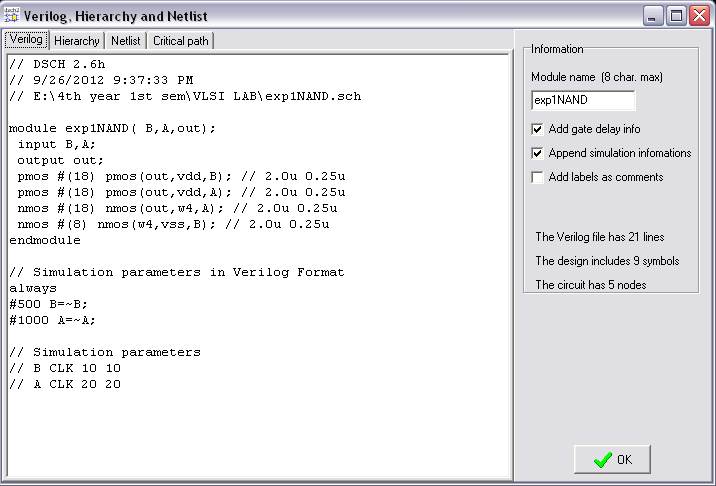


Fig: Timing Diagram

### Circuit design simulation:

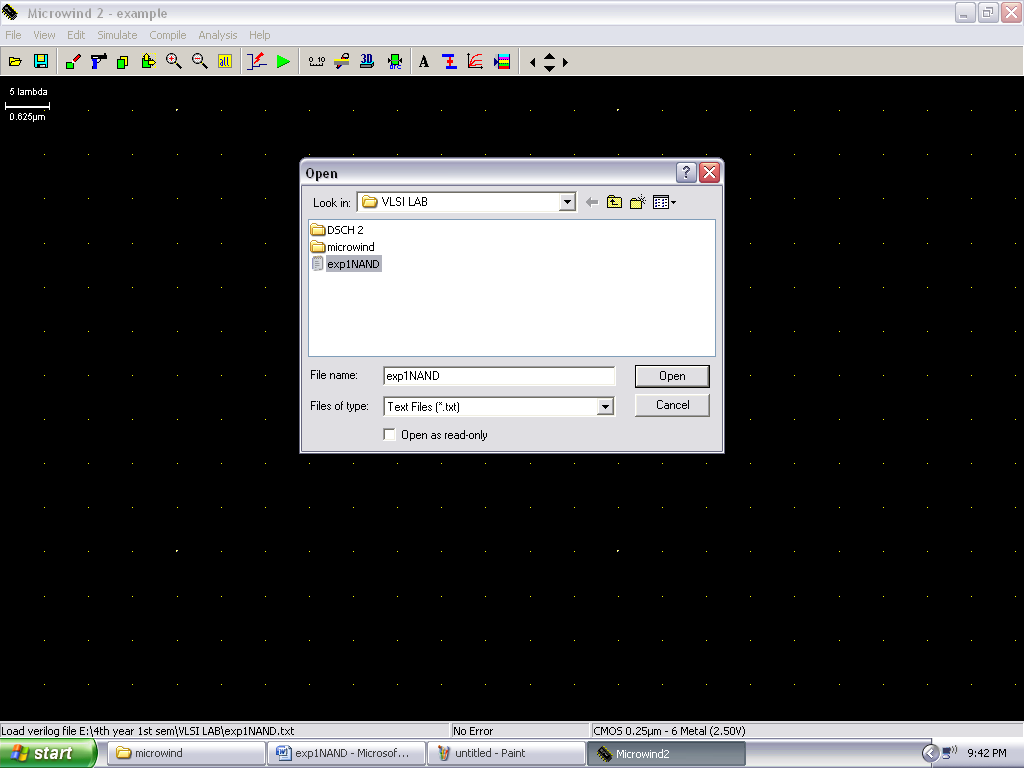
* Open the schematic that we created in the logic design stage. The next step is to convert the schematic into a Verilog representation. Click on *File🡪 make Verilog File*. The Verilog, Hierarchy and Netlist window will appear. This window shows the verilog representation of our (NAND gate in this example) circuit. Click on *Ok*. Thes saves the verilog file as ‘.txt’ file in the present directory. Our circuit is now saves as a Verilog file. Close the schematic editor window.



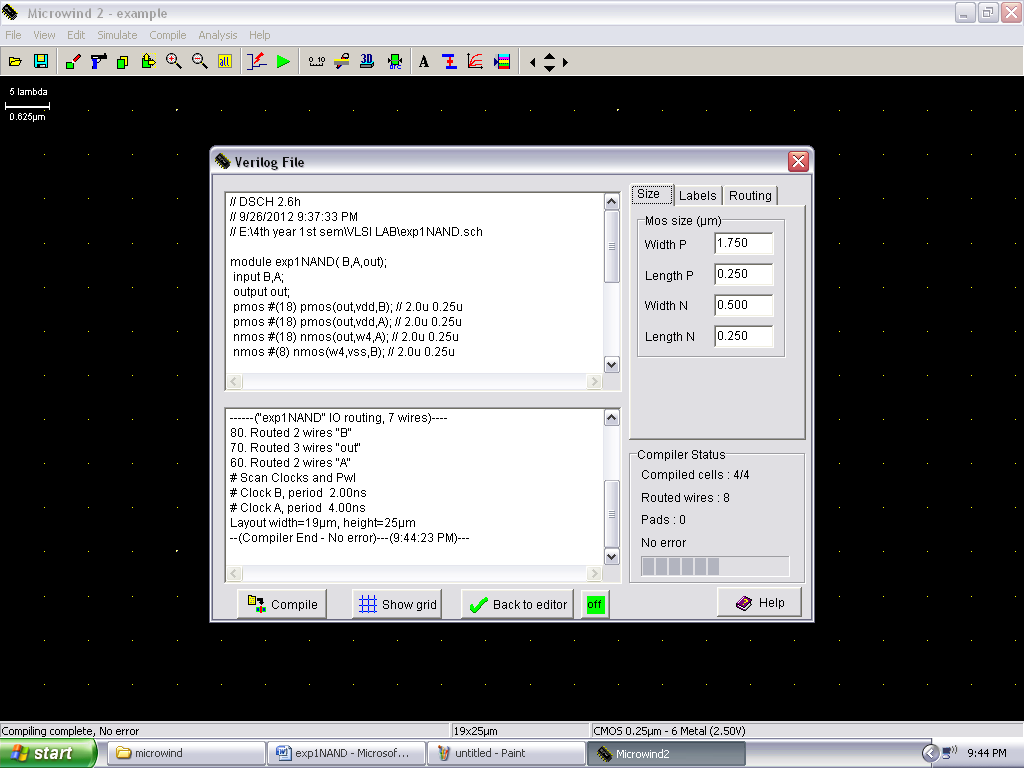


Fig; Verilog Code

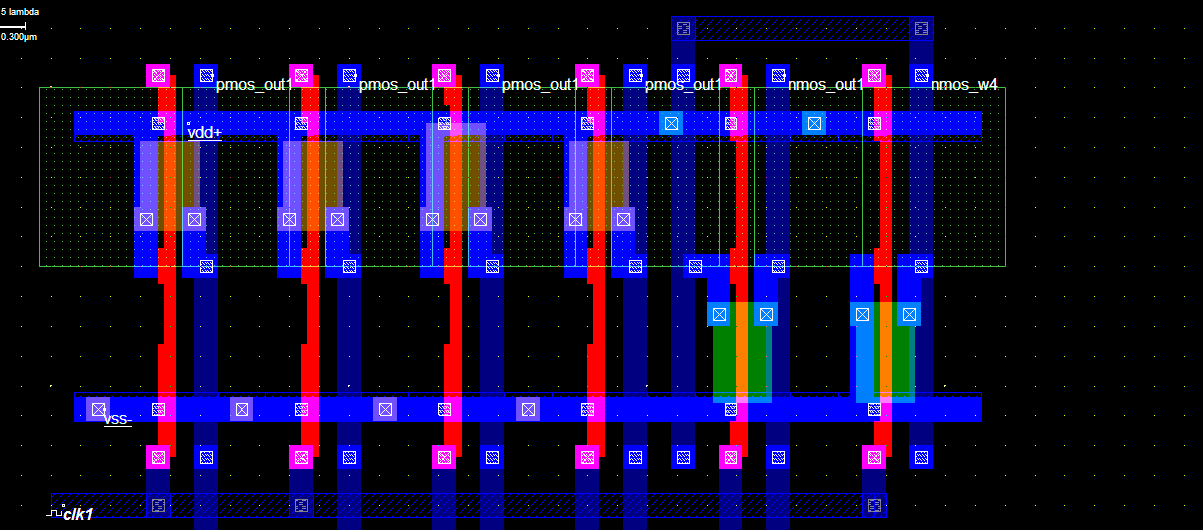
* Now open the layout editor window ‘Microwind2’. Click on *File🡪Select Foundry* and select *cmos025.rul*.
* Click on *Compile🡪Compile Verilog File*. An window will appear shown below and select the verilog file as you saved before. Open it.

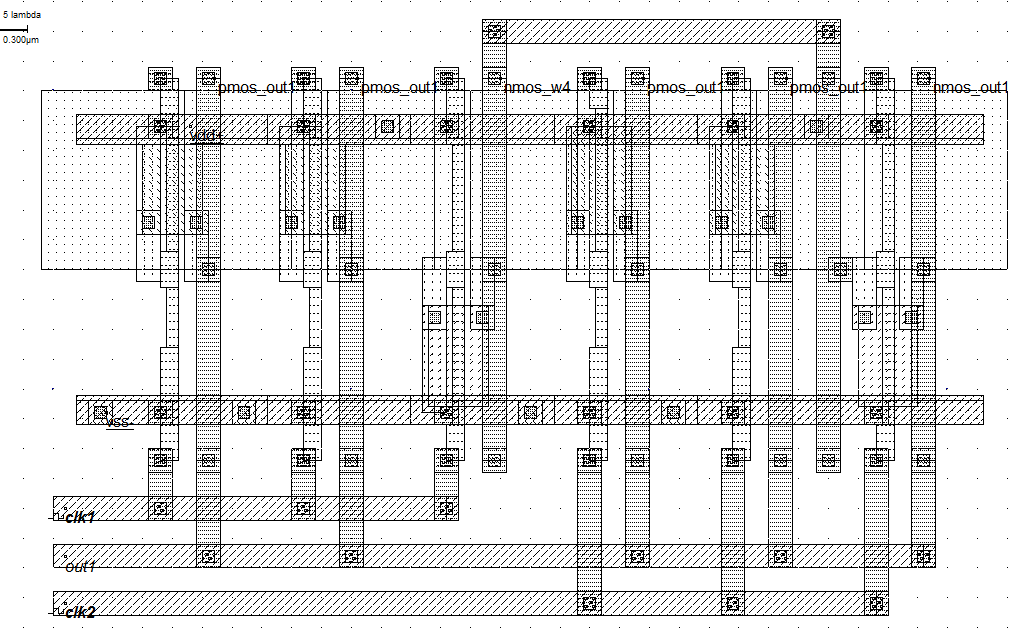


* A new window will appear. Click on the *Size*  tab then it will display the size of the nMOS and pMOS gate. We can set the size of the nMOS and pMOS in our design.

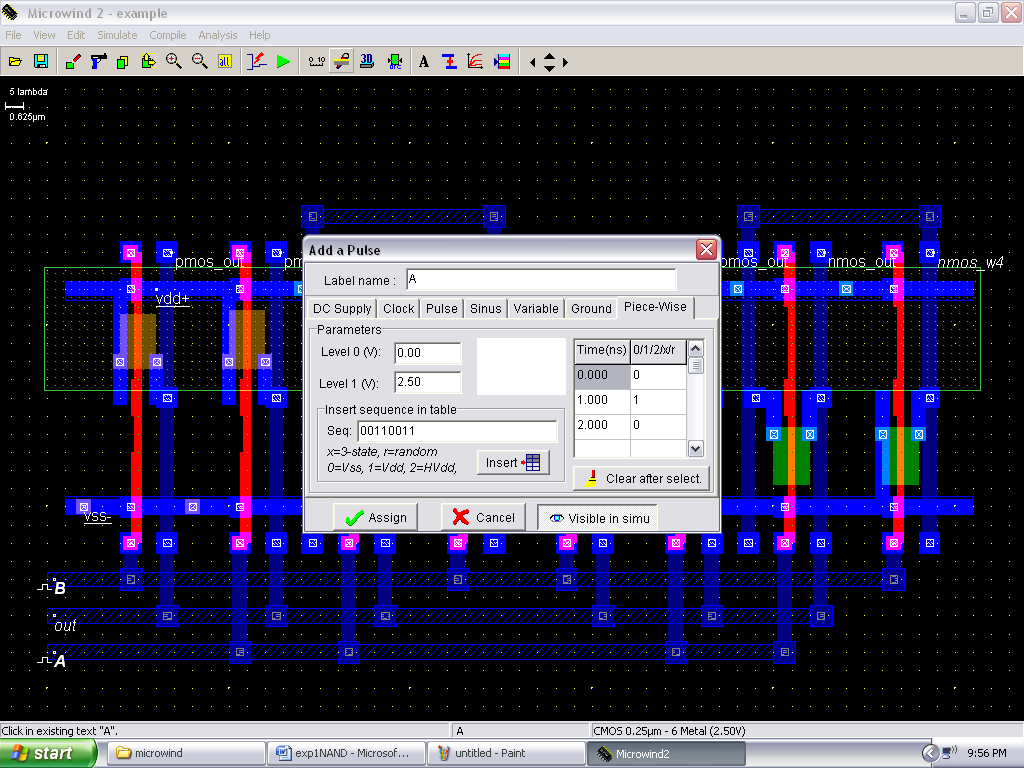


* Now click on compile button. It will compile the verilog file and finally it shows the no error if successfully complied the code. Click on 🗶 to show the automatic generation of layout design in our design show below:

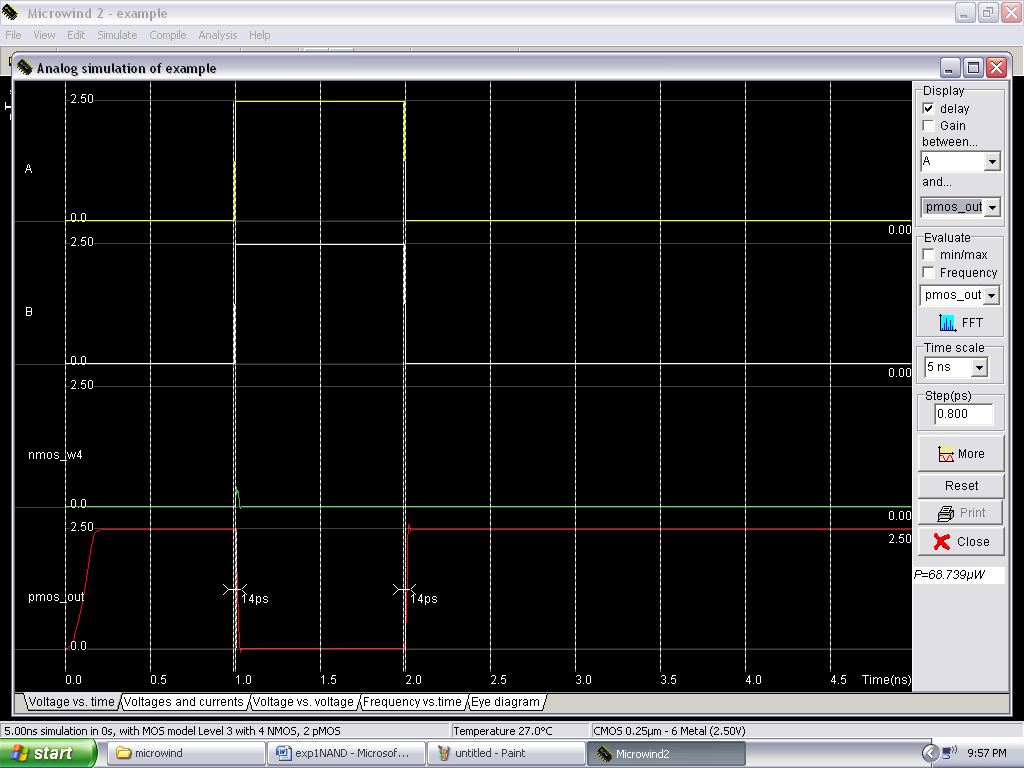




* Click on the label marked ‘in1’. A window will appears shown below. Click on the Piece-Wise option in the window. Insert a 01 sequence for that specific input and click on insert. Then click on Assign. Perform this assignment on the other inputs also.



* Click on Simulate🡪Run Simulation. A simulation will appears with the inputs and output being shown. Measure the tphl, tplh and tp of the circuit using this simulation window. You can click and drag in the simulation window in the hozizontal direction to make the measurement. Rising and falling delays are shown on the output waveform. The power consumption is also shown on the right bottom portion of the window.



* If you are unable to meet the specifications of the circuit as given in the laboratory description, change the size of nPOS and pMOS. Generate the layout again and run the simulation till you achieve your target delays.

### Conclusion:

The NAND gate is implemented using two pMOS and two nMMOS and the truth table is successfully verified. The required waveforms were obtained, observed and noted down using microwind2.