

EAMTA  
2024

Argentine School of  
Micro-nano Electronics,  
Technology and Applications

## Basic-VLSI

Agustín Galetto - Nicolás Calarco

Last edited February 29, 2024



From March 04 to 08, 2024  
UNS, Bahía Blanca, Argentina  
[eamta.ar@gmail.com](mailto:eamta.ar@gmail.com)  
[eamta.ar](http://eamta.ar)



## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>CAD tools environment installation and usage</b>	<b>2</b>
<b>3</b>	<b>Schematic capture of a CMOS inverter</b>	<b>3</b>
3.1	Runing Xschem . . . . .	4
3.2	Schematic creation . . . . .	4
<b>4</b>	<b>Inverter symbol creation</b>	<b>6</b>
<b>5</b>	<b>Inverter test-bench creation and simulation</b>	<b>7</b>
5.1	Code used into some components . . . . .	8
5.2	Running the simulation . . . . .	9

# 1 Introduction

The goal of the course is for the student to understand the physical fundamentals of MOS device operation and the basic principles of CMOS integrated circuit fabrication. Also, for the student to acquire the ability to design basic analog and digital circuits and use CAD tools.

The course is based on the use of the Open-source PDK provided by skywater and the Open-source EDA Tools. See [https://efabless.com/open\\_shuttle\\_program](https://efabless.com/open_shuttle_program).

In order to facilitate the installation process and ensure the correct functionality of the tools, a Docker container with the complete environment installed it will be provided. This option is provided by the Institute for Integrated Circuits (IIC), at Johannes Kepler University (JKU). It is Open Source and has a large community of collaborators. You can find the repository on <https://github.com/iic-jku/IIC-OSIC-TOOLS>.

As a hello world example, in this tutorial, we are going to cover the whole design flow which comprehends:

- Installation of the CAD tools.
- Schematic capture of a CMOS inverter.
- Inverter symbol generation.
- Test-bench generation to evaluate the inverter performance.
- Clean DRC and LVS Inverter layout design.
- Layout parasitic extraction.
- Post-layout inverter re-simulation.

## 2 CAD tools environment installation and usage

The usage instructions can be founded in this git repository: <https://github.com/iic-jku/IIC-OSIC-TOOLS>. In this section we will try to simplify the process described in the section How to Use These Open-Source (and Free) IC Design Tools, but please refer to the repository in case of doubts or troubles.

Here are enlisted the steps for installing the environment:

1. **Download or clone the repository** <https://github.com/iic-jku/IIC-OSIC-TOOLS>.  
In this folder you have everything you need for running the tools. Now is necessary to install some additional software to execute correctly the environment. **Note:** This project is continually updated, solving bugs or improving functionalities; if you download the repository, you will not be able to update it.
2. **Install Docker.** In the Quick Launch for Designers section are posted the installation files for all Operating Systems. Choose yours, download and install Docker on your PC. Docker website: <https://www.docker.com/>. **Note:** For correct execution of Docker, the *Hardware Virtualization* function must be activated in your PC. This function can be already activated, but if not, you probably get an error like *Windows hipervisor is not present* when you try to run Docker. Please search in Google how to enable Hardware Virtualization in your specific PC. If the problem persist you can find some guidance [HERE](#).
3. **Install an X11 server.** We are going to use an X11 interface. In this case, for Windows users, an X11 server is required. Every X11-server should work, although those that appear in the section Installing X11-Server are tested. After installation, an **XLaunch** application should appear in your PC. Each time you use the tools, you have to launch this application before. Please follow the configuration recommendations posted in the section when you use it:

On Windows with VcXsrv, we recommend using the utility "XLaunch" (installed with VcXsrv):

- Multiple windows mode
- Set Display Number to 0
- Start no client
- **Tick all** Extra settings: Clipboard, Primary selection, Native opengl, and Disable access control

4. **Setup your environment.** Go to your repository folder and execute the "x" option: *start\_x.bat* for Windows, or *start\_x.sh* for Linux users. This will create some environment variables and the container in the Docker application.

At this point, you have all the components installed. Each time you want to run the environment please:

- **First**, execute the **XLaunch** application (see recommended configuration in item 3 of Section 2).
- Once XLaunch is running in background (you will see the Xlaunch icon in your system tray), open **Docker** and run the iic-osic-tools container.

A terminal window will appear and you will be able to run the tools.

**Note 1:** If you do this the first time, or an updated image to DockerHub has been pushed, this can take a while since the image is pulled (loaded) automatically from DockerHub. Since this image is ca. 4GB, this takes time, depending on your internet speed. Please note that this compressed image will be extracted on your drive, so please provide at least 20GB of free drive space. If, after a while, the consumed space gets larger, this is maybe due to unused images piling up. In this case, delete old ones; please consult the internet for instructions on operating Docker.

**Note 2:** All user data is persistently placed in the directory pointed to by the environment variable DESIGNS. The default locations are:

- **Windows:** %USERPROFILE%\eda\designs
- **Linux:** \$HOME/eda/designs

We will refer to this locations as **designs folder** from now.

### 3 Schematic capture of a CMOS inverter

Xschem is an open-source schematic capture software for designing electrical circuits. It is used for creating schematics, which are graphical representations of an electronic circuit.

Some of the key features of Xschem include:

1. Component Library: Xschem has a large library of components that can be easily placed onto the schematic canvas to create the circuit.
2. Wiring: Xschem has an easy-to-use wiring tool that makes it simple to connect components together to form a complete circuit.
3. Annotation: Xschem allows you to add labels and net names to the schematic to help identify different signals in the circuit.
4. Simulation: Xschem can export the schematic to other simulation tools, such as ngspice, for analysis and testing.

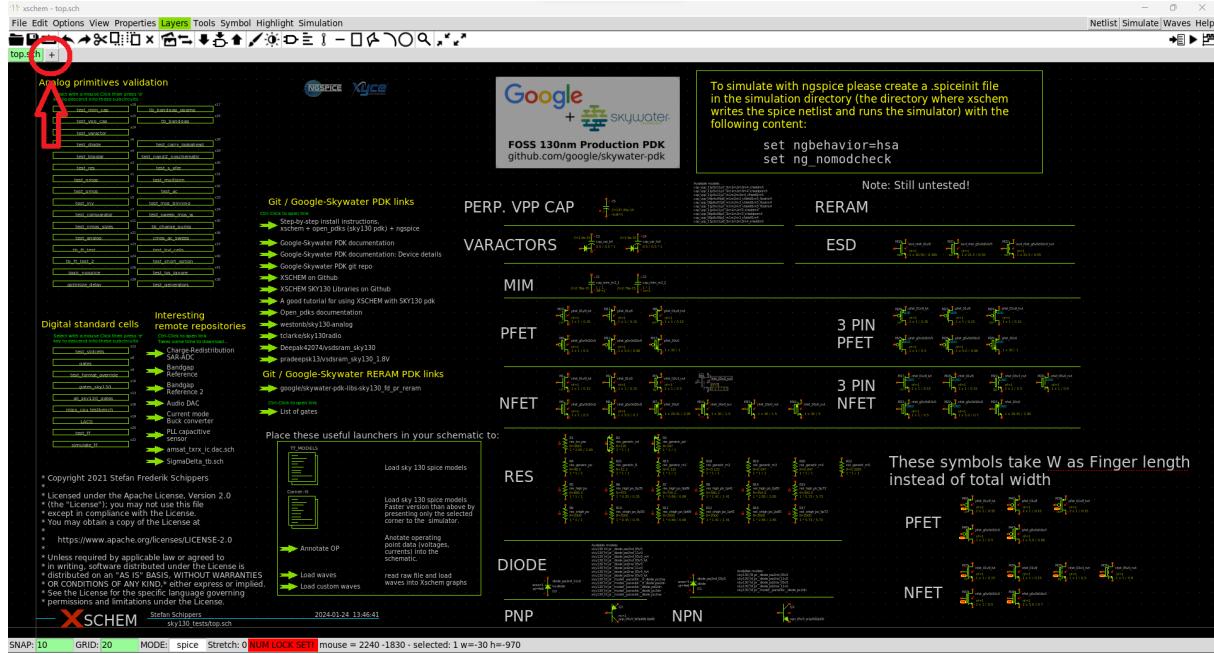


Figure 1: Welcome screen when Xschem is opened.

5. Cross-platform: Xschem runs on multiple operating systems, including Linux, macOS, and Windows.
6. Customization: Xschem allows users to create custom components and macros to streamline the schematic capture process.

This is a general overview of the Xschem schematic capture software. It is a powerful tool that can be used for a wide range of electronic circuit design tasks. The Xschem manual and online resources can provide more detailed information and guidance.

### 3.1 Runing Xschem

Once the Container is launched, to run *Xschem* go to the terminal window and write:

```
xschem
```

A sort of welcome screen like shown in Fig. will appear. Yo will see here many components and basics designs that are already included in the repository.

### 3.2 Schematic creation

In this section you will generate the schematic of the inverter shown in Fig. 2.

Xschem is a common schematic design software such as any other, so we assume you are familiarized with. We will give just a couple of instructions regarding components and paths. You can find a brief resume of the main functionalities shortcuts in Table 1.

Please follow the next recommendations for your new design:

- Open a new tab by clicking in the ‘+’ symbol indicated by the red arrow on Fig. 1. Click on File -> Save as and save this new design as inverter.sch, ensuring that you are in the folder indicated by a circle. This folder (/foss/designs) points to the **designs folder** defined on section 2. Then, you can go quickly to this path by clicking **Current dir** in the saving or instancing windows.

Key	Function
u	Undo
U (Shift+u)	Redo
Left-click.	Select element
Double-click. / q	Edit. element prop.
scroll	Zoom in/out
f	Center Zoom
w	Create wire
shift+i , ins	Insert component
c	Copy
m	Move
shift + f	Mirror
shift + r	Rotate
Esc	Release function
Supr/delet	Delete

Table 1: Xschem Shortcuts

- For instancing components pressing the **insert** key. **Please, pay attention on the path:** transistors are encountered into the library `/foss/pdks/sky130A/libs.tech/xschem/sky130_fd_pr`, which we will name from now as the **technology folder**. The specific components to choose are **nfet\_01v8.sym** and **pfet\_01v8.sym** for the *nmos* and *pmos* respectively. See Fig. 3 for reference. **Note:** Be careful NOT TO SELECT the three terminals devices **pfet3** or **nfet3**!
- To make some **edits**, note that a pre-selection of the element with a left click is required (See Table 1). Set the device properties to those shown in the Fig. 2 (only a change in the **w** parameter is needed).
- To create **pins** or **labels**, use any of the insert component shortcuts and go to the xschem library. You can find this library written in red when you press the Home button. Enter to **devices** and select *iopin.sym* and *lab\_pin.sym* respectively for pins and labels, like is shown in Fig. 4.

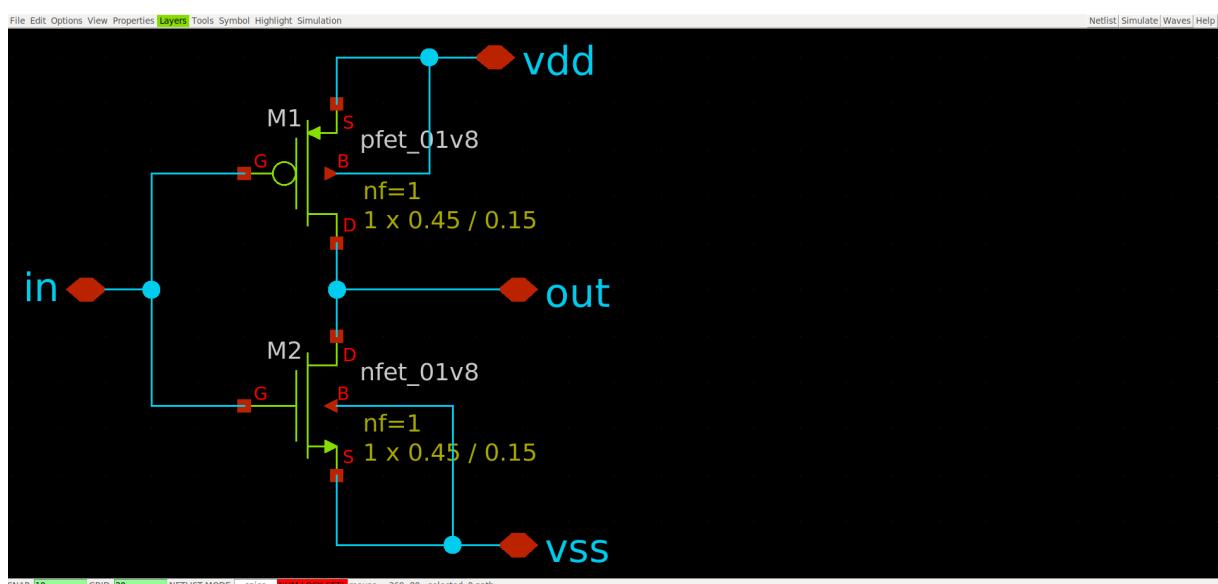


Figure 2: Inverter final schematic.

## 4 Inverter symbol creation

Once you obtain the design of Fig. 2, generate the symbol of the inverter shown in Fig. 5.

For this, first generate a symbol for this schematic, by pressing **A** or clicking on Symbol -> Make symbol from schematic, as is shown in Fig. 6.

A new file named **inverter.sym** will be generated in the same folder than the schematic. Open it (open a new tab, click File -> Open and select **inverter.sym**) and try to obtain the symbol of the Fig. 5. The symbol editor is different than the schematic editor, because it allows to modify shapes and names. To make some edits, note that a pre-selection of the element with a left click is required. The shortcuts indicated in the table 2 can be taken into account; for example, for drawing a line, press **l** in your keyboard.

Key	Function
u	Undo
U (Shift+u)	Redo
Left-click.	Select element
Double-click. / q	Edit. element prop.
scroll	Zoom in/out
f	Center Zoom
l	Line
c	Copy
m	Move
shift + f	Mirror
shift + r	Rotate
Esc	Release function
Supr/delet	Delete

Table 2: Xschem Shortcuts (Symbol view)

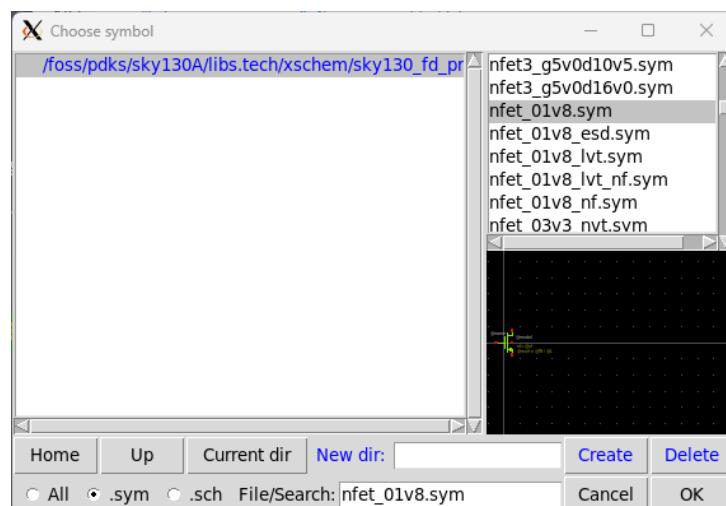


Figure 3: Path and component for an nmos transistor instantiation

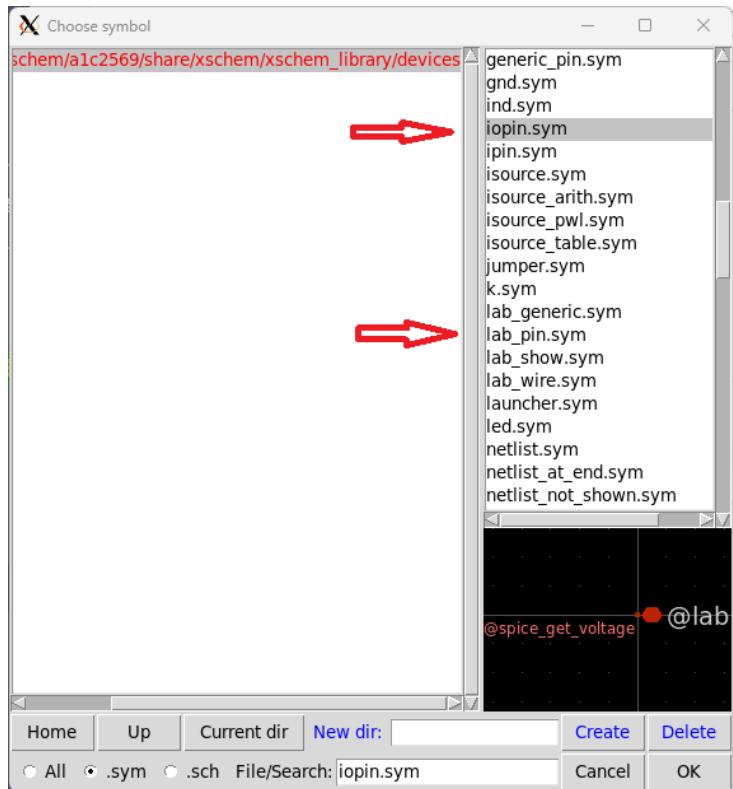


Figure 4: Path and component for pins and labels instantiation

## 5 Inverter test-bench creation and simulation

To simulate the inverter we will use the circuit the test-bench circuit shown in Fig. 7. This circuit has the inverter as unit under test (**uut**), several stimulus and an inverter load (this is a typical method to probe the components). An additional capacitor is added just for connecting the last node. So, We are going to create new schematic, which we will save as `tb_inverter.sch`, then instance the components and make the connections. Note that the schematic has four components, which can be found in three different paths:

- **The inverters** (the **uut**), which are placed in your designs folder. You can access it quickly clicking on the **Current dir** button.
- **The power sources**, V4, V5 and V6, located on the xschem library. You can access it quickly clicking on the **Home** button and selecting the **device** folder of the xschem library (marked in red). All the of them are **vsource** components with different values that are showed in Section 5.1.

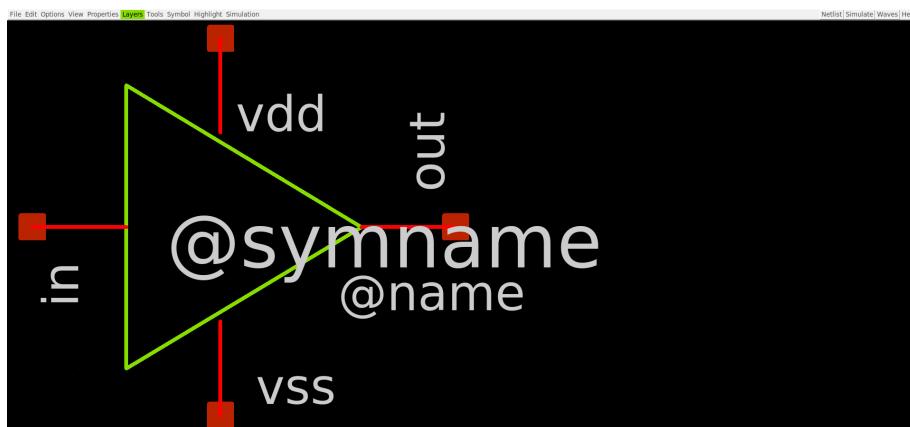


Figure 5: Inverter final symbol.

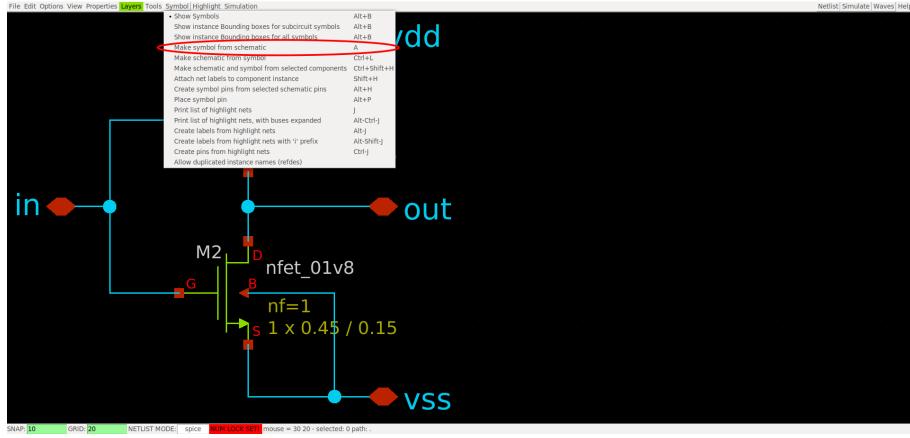


Figure 6: Instructions to create a symbol from the inverter.

- The **netlist \_not \_shown.sym** component is in which the simulations will be described. Its content will be described in Section 5.1
- The **capacitor**, that can be found under the **xschem folder** clicking on the **Home** button and then selecting the **technology folder** (see section 3.2). Select the *capa.sym* component.
- The **GND symbol**, also located on the xschem library.

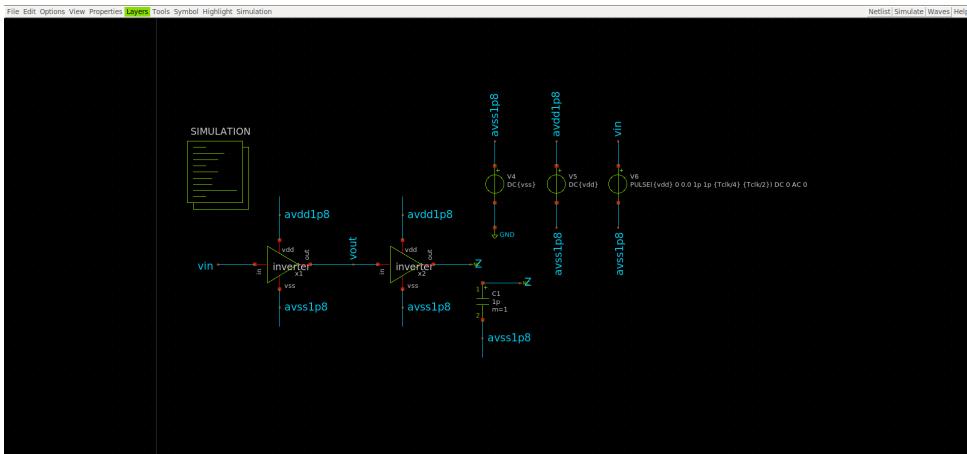


Figure 7: TB Inverter schematic.

## 5.1 Code used into some components

To insert the necessary code for each component, please double-click or pre-select (left-click) and press *q*. All the values in braces are parameters that can be changed by the simulation definition.

V4 and V5 are DC sources so its value has to be changed to:

```
value=DC{vdd}
```

V6 is a pulse source so its value has to be changed to:

```
value="PULSE({vdd} 0 0.0 1p 1p {Tclk/4} {Tclk/2}) DC 0 AC 0"
```

**Note:** Please respect the name of **V6** in the testbench, because it will be used after in the simulation script!

Code used into *netlist \_not \_shown.sym* (you can erase all its content and paste this)

```

name=SIMULATION only_toplevel=false
value=""

* Circuit Parameters
.param vdd = 1.8
.param vss = 0.0
.param Tclk = 10n
.options TEMP = 65.0

* Include Models
.lib /foss/pdks/sky130A/libs.tech/ngspice/sky130.lib.spice TT

* OP Parameters & Signals to save
.save all

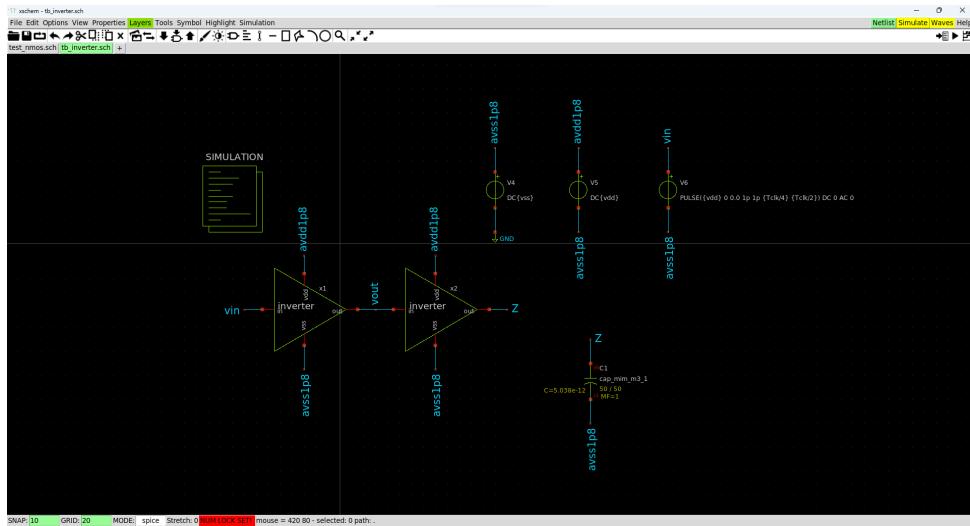
*Simulations
.control
  tran 0.01u 100n
  setplot tran1
  plot v(vin) v(vout)+2
reset
  dc V6 0 1.8 0.01
  setplot dc
  plot vin vout ylabel vout xlabel vin
  set filetype = ascii
  write dcsweep.raw
.endc

.end
"

```

## 5.2 Running the simulation

To run the simulations, click first in the Netlist button (indicated in green on the right-upper corner of the screen in Fig. 8), and then Simulate (next to the Netlist button, indicated in yellow in Fig. 8). This process is **always** necessary after any change: first Netlist and then Simulate.



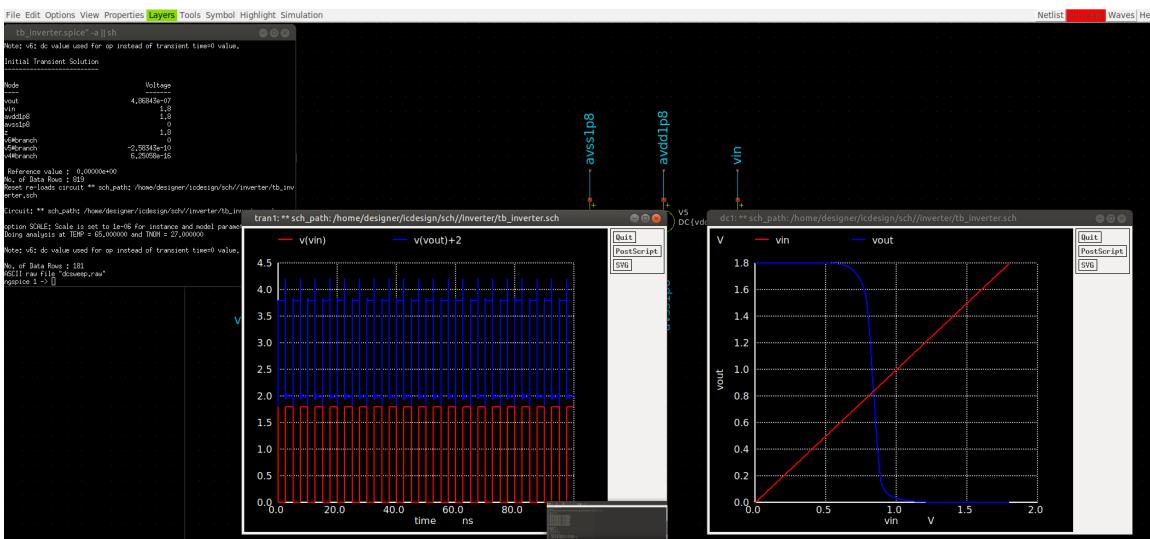


Figure 9: TB Inverter results.