Circuit Integration Internship (PSI)

Experiment No.: **SI2**
Experiment title: **Dynamic behavior of CMOS inverters and gates**
Experimental concept: **C. Schimpfle / F. Aschauer / D. Kohlert**

I. Experimental objective

In experiments SI0 to SI3, you will learn how to use a very powerful EDA (Electronic Design Automation) software package used in industry. You will practice using the programs using complete design flows for fully custom and standard cell designs.

II. Brief description of the experiment

In this experiment, you will learn the complete design process from schematic entry to post-layout analysis. To practice hierarchical design, you will first design a quadruple NAND gate as a standard cell. You will then combine this with existing standard cells to create a (slightly) more complex functional block. You will examine this and other existing functional blocks with regard to their timing behavior.

III. The design flow with Cadence EDA programs

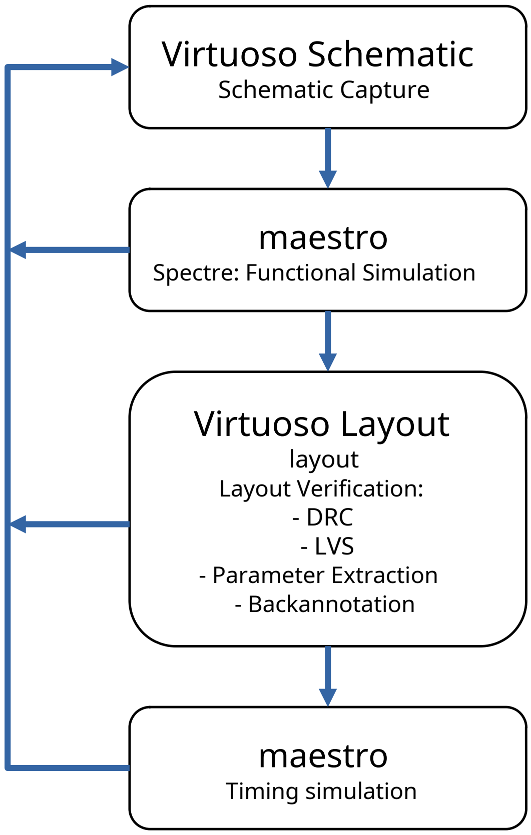


Fig. 1: Schematic-based design workflow

Figure 1 shows a schematic-based design workflow with Cadence. It includes the following steps:

- Schematic capture with**VIRTUOSO**. You can also use this program to assign a symbol of your choice to any collection of linked components in order to then reuse this functional block at a higher hierarchy level.

- Functional simulation with**SPECTRE**This is where we check whether the circuit has the desired functionality. A successful simulation is usually followed by test vector generation, i.e., the definition of the signals (stimuli) with which the produced circuits will be tested (see also the PTT practical course).

- Layout creation with**VIRTUOSO XL**In this step, the geometry data is generated with which the masks for the production of the integrated circuit are manufactured. Depending on the hierarchy level, individual polygons are edited (lowest level) and thus, for example, standard cells are designed, or (standard) cells are placed and wired, so that more complex functional blocks are created. The program also performs important tests and controls for*Layout verification*by: checking whether all geometric design rules have been complied with (**Design Rule Check, DRC**), checking whether the layout topologically matches the circuit diagram (**Layout Versus Schematic, LVS**) and finally the extraction of component parameters for the subsequent (analog) simulation of the time behavior.

- Post-layout simulation with**SPECTRE**. Using the extracted parameters for the transistors and parasitic capacitances, an analog (SPICE) simulation is performed to verify the (time) behavior of the circuit.

A program package like the one just described only becomes usable when it is "fed" with appropriate technology data. In this lab, we will use the Skywater 150nm Design Rules.

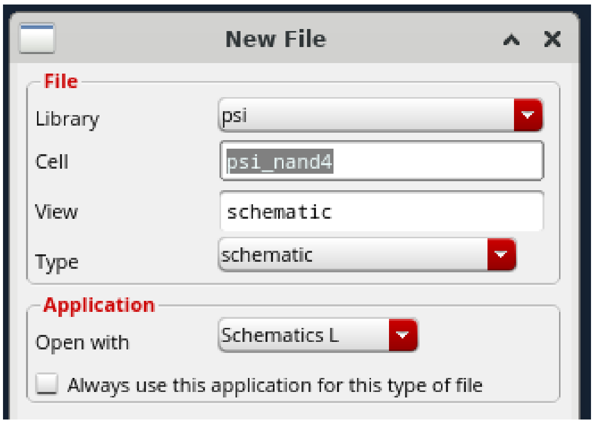
IV. Experimental procedure

IV.1 Schematic input for the 4-way NAND gate

Draw a 4-way NAND gate circuit diagram:
Call Virtuoso
In the Virtuoso Command Window: File/New/Cellview

Window appears:
Fill out as shown

"OK"



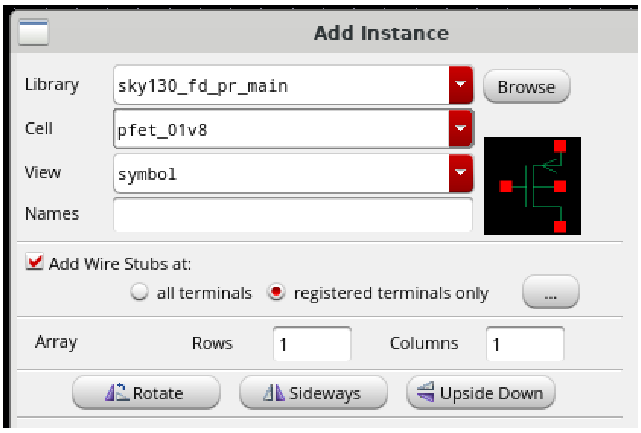
New File

File
Library: psi
Cell: psi_nand4
View: schematic
Type: schematic

Application
Open with: Schematics L
□ Always use this application for this type of file

Drawing window appears

Get transistor from library:
Keyboard shortcut: "i":
Window appears:
Fill out as shown:
p-FET from library sky130_fd_pr_main is inserted:



Add Instance

Library: sky130_fd_pr_main
Cell: pfet_01v8
View: symbol
Names:
☑ Add Wire Stubs at: ○ all terminals ● registered terminals only
Array Rows: 1 Columns: 1

Symbol hangs on the cursor, can be placed with LMB.

Below a window with transistor properties appears, can remain as it is.

Insert n-FET in the same way

Copying components: Keyboard shortcut: "c":
Move component:
Mouse pointer on component until yellow rectangle appears, move with LMB pressed

Wiring components: Keyboard shortcut: "w"
Place external connector (pins): Keyboard Shortcut: "p", window is self-explanatory

Create the circuit of a 4-way NAND

Place all inputs, outputs and also supplies as pins. Supply connections are "Input", attention to output, must be "Output"

To avoid problems with keywords (in, out, vdd, vss, etc.), port names should be chosen that are definitely not keywords.
Suggestion: All names begin with "n_" ("n" for node)

With "Check and Save" errors in the circuit are detected, carry out occasionally, but always at the end!

IV.2 Functional Simulation

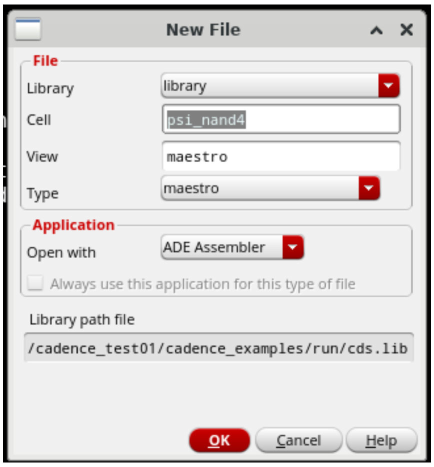
IV.2.1 Preliminary considerations

The simulation is initially intended only to prove the functional correctness of the gate; investigations of the runtime behavior will be carried out later, when the influences of parasitic capacitances can be taken into account in the layout.
It is therefore simulated here with the unloaded gate.

For the simulation, consider stimulus sequences for testing the NAND (4 sources), covering all possible input bit combinations. Due to the expected gate delays, a pulse width of at least 500 ps should be used.
The supply voltage for this technology is 1.8V. Document the
signal curves!

IV.2.2 Setup Simulation

cite sources
In the Virtuoso Command Window:
Start/New/Cellview
The window shown below appears
Specify type "Maestro"
Cell: name of the cell to be simulated (here: psi_nand4) ok



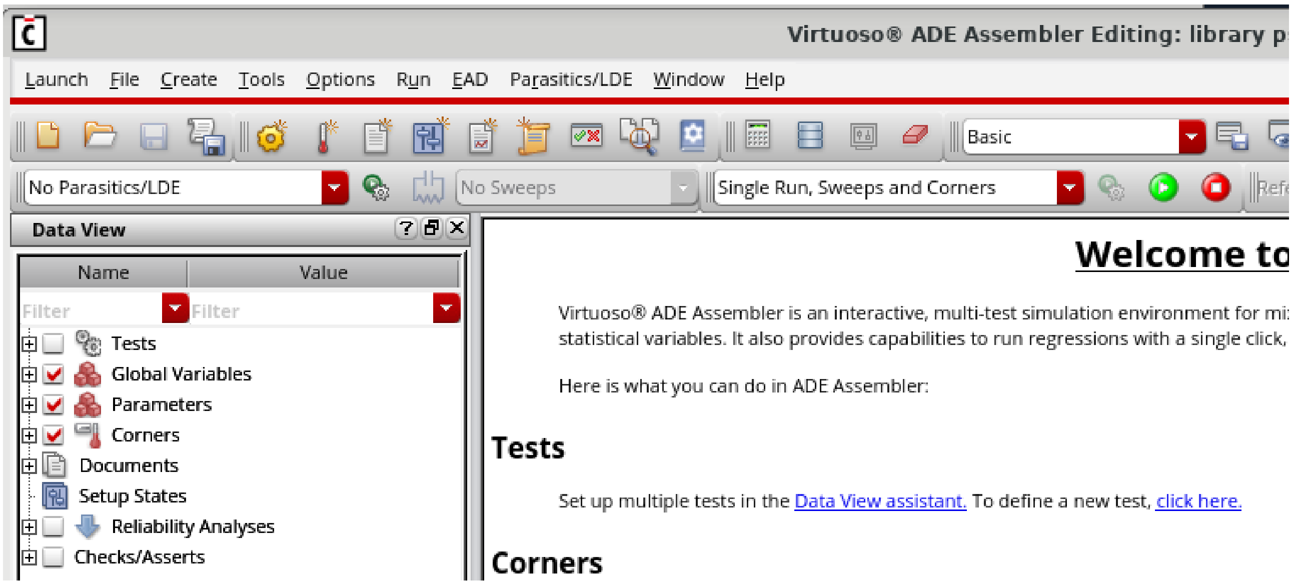
New File

File
Library: library
Cell: psi_nand4
View: maestro
Type: maestro

Application
Open with: ADE Assembler
☐ Always use this application for this type of file

Library path file
/cadence_test01/cadence_examples/run/cds.lib

Window appears:



Virtuoso® ADE Assembler Editing: library p

Welcome to

Virtuoso® ADE Assembler is an interactive, multi-test simulation environment for mi
statistical variables. It also provides capabilities to run regressions with a single click,

Here is what you can do in ADE Assembler:

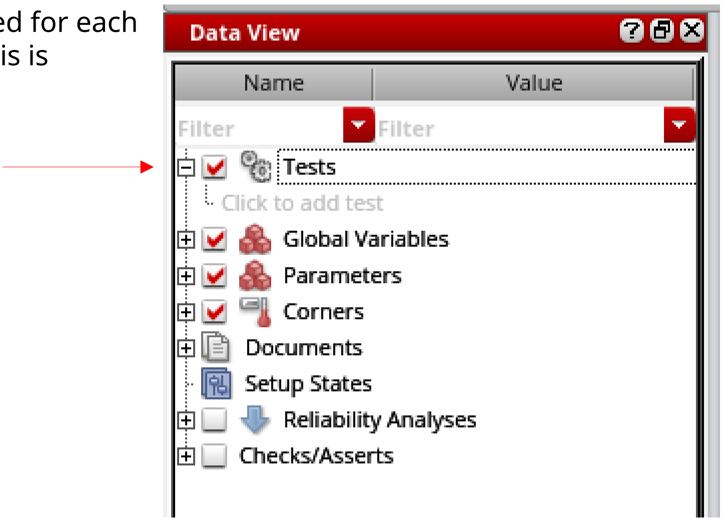
Tests

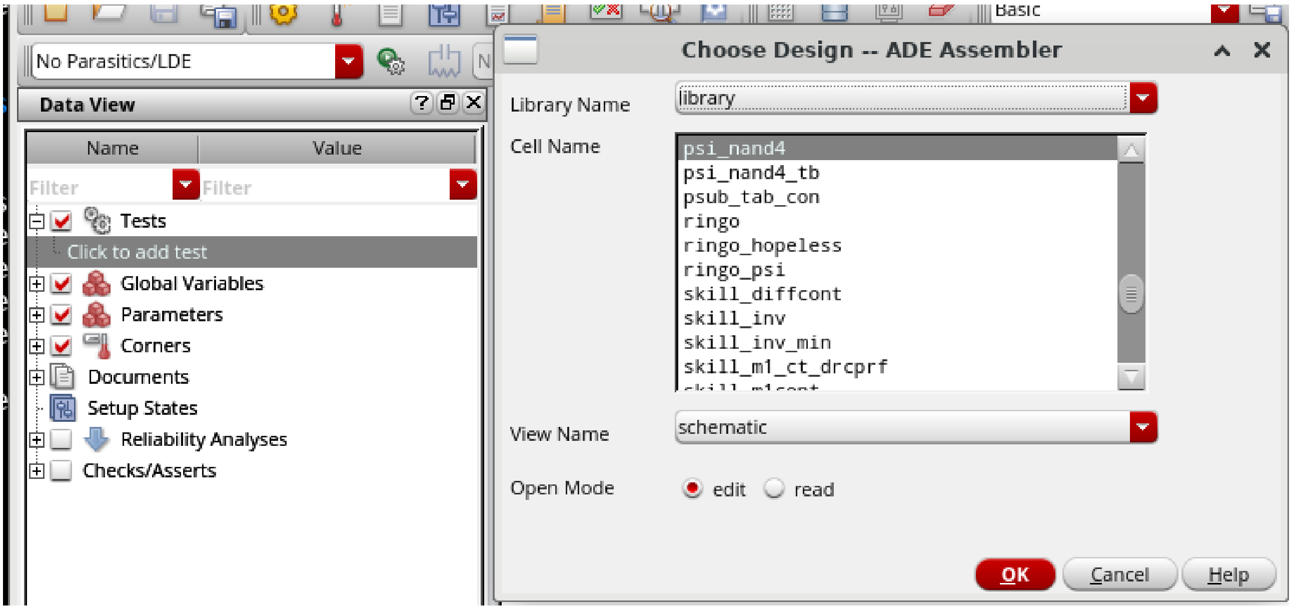
Set up multiple tests in the Data View assistant. To define a new test, click here.

Corners

Create new test:

Any number of different tests can be created for each cell, but here one test with transient analysis is sufficient.





Library Name: library

Cell Name:

```
psi_nand4
psi_nand4_tb
psub_tab_con
ringo
ringo_hopeless
ringo_psi
skill_diffcont
skill_inv
skill_inv_min
skill_m1_ct_drcprf
```

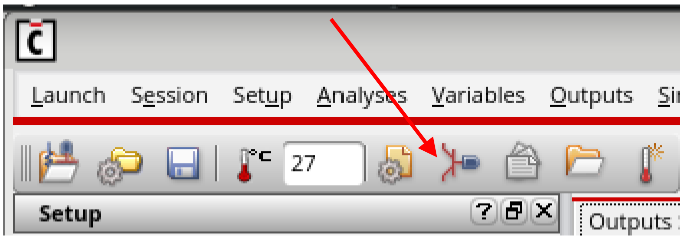
View Name: schematic

Open Mode: ● edit ○ read

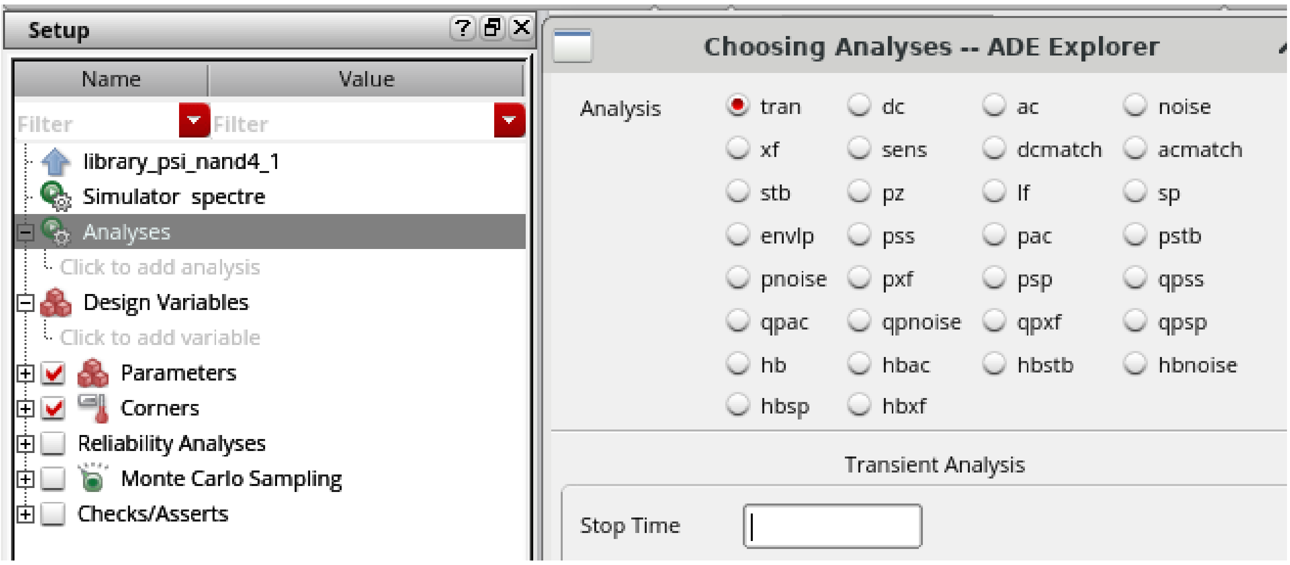
"Click to add test" -> selection window appears, select desired cell

Empty window appears

Indicate sources: Click transistor symbol:



Select analysis mode:



Setup

Name	Value
Filter	Filter
library_psi_nand4_1	
Simulator spectre	
Analyses	
Click to add analysis	
Design Variables	
Click to add variable	
☑ Parameters	
☑ Corners	
□ Reliability Analyses	
□ Monte Carlo Sampling	
□ Checks/Asserts	

Choosing Analyses -- ADE Explorer

Analysis				
	● tran	○ dc	○ ac	○ noise
	○ xf	○ sens	○ dcmatch	○ acmatch
	○ stb	○ pz	○ lf	○ sp
	○ envlp	○ pss	○ pac	○ pstb
	○ pnoise	○ pxf	○ psp	○ qpss
	○ qpac	○ qpnoise	○ qpxf	○ qpsp
	○ hb	○ hbac	○ hbstb	○ hbnoise
	○ hbsp	○ hbxf		

Transient Analysis

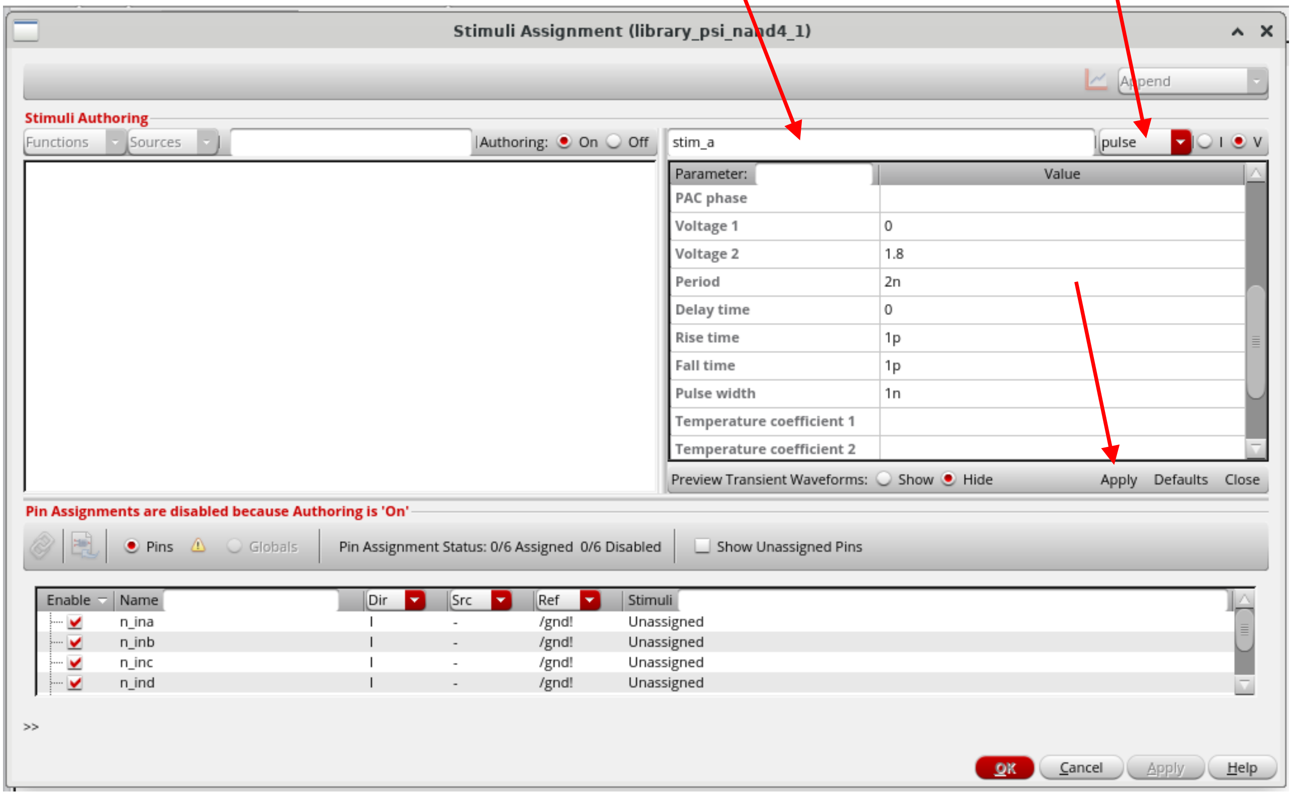
Stop Time

Here: Select transient analysis Specify stop time (e.g. 1nNOT ns!!)

Stimuli assignment window appears, IO's already appear in the lower window: If not, click transistor symbol again
Assign sources: "Authoring" mode must be selected Source names are not the same as signal names

Name of source (write in):

Source:



Stimuli Assignment (library_psi_nand4_1)

Stimuli Authoring

Functions | Sources | Authoring: ● On ○ Off | stim_a | pulse | ○ I ● V

Parameter:	Value
PAC phase	
Voltage 1	0
Voltage 2	1.8
Period	2n
Delay time	0
Rise time	1p
Fall time	1p
Pulse width	1n
Temperature coefficient 1	
Temperature coefficient 2	

Preview Transient Waveforms: ○ Show ● Hide

Pin Assignments are disabled because Authoring is 'On'

● Pins ○ Globals Pin Assignment Status: 0/6 Assigned 0/6 Disabled □ Show Unassigned Pins

Enable	Name	Dir	Src	Ref	Stimuli
☑	n_ina	I	-	/gnd!	Unassigned
☑	n_inb	I	-	/gnd!	Unassigned
☑	n_inc	I	-	/gnd!	Unassigned
☑	n_ind	I	-	/gnd!	Unassigned

Type of source: Here "pulse", all common spice sources are available

Rectangular signals with a minimum pulse width of 500ps, rise time = fall time = 1p and 50% duty cycle should be used.

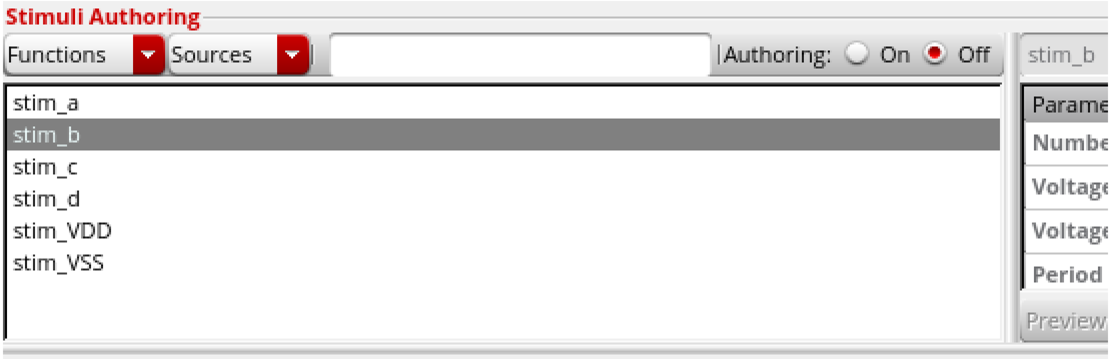
Voltage 1: Low level
Voltage 2: High level (1.8V)
Period: Period duration
Delay Time: Waiting time at the beginning of the simulation (0ps
here) Rise Time: Rise time of the signal (e.g. 1ps)
Fall Time: Fall time of the signal (e.g. 1ps)
Pulse Width: Pulse width

When all entries are made: press "Apply", the information will be adopted

Provide information for all sources, don't forget DC sources!

If all sources are described: Turn off "Authoring" mode

Assigning the stimuli:
Select signal in the authoring window

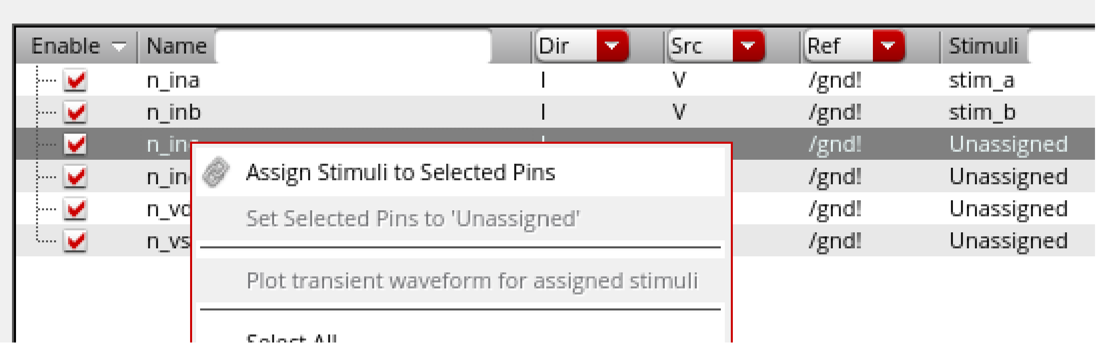


Stimuli Authoring

Authoring: ○ On ● Off

stim_a
stim_b
stim_c
stim_d
stim_VDD
stim_VSS

Right



Enable	Name	Dir	Src	Ref	Stimuli
☑	n_ina	I	V	/gnd!	stim_a
☑	n_inb	I	V	/gnd!	stim_b
☑	n_in	I		/gnd!	Unassigned
☑	n_in			/gnd!	Unassigned
☑	n_vd			/gnd!	Unassigned
☑	n_vs			/gnd!	Unassigned

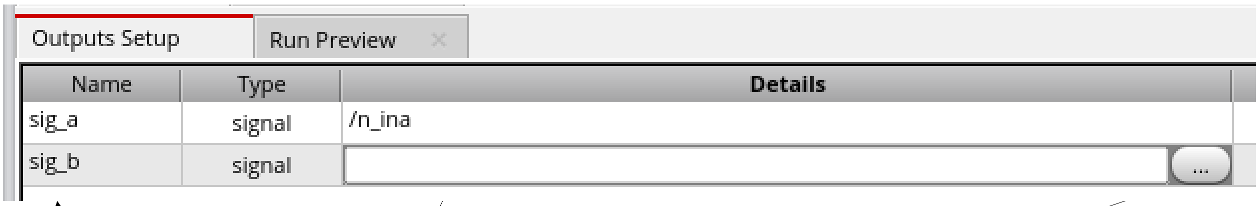
Assign Stimuli to Selected Pins
Set Selected Pins to 'Unassigned'
Plot transient waveform for assigned stimuli
Select All

When everything is done: Apply/ok

Plotting signals:

Select signals:

Main menu: Outputs/Add/Signal
window appears:



Name	Type	Details
sig_a	signal	/n_ina
sig_b	signal	

Any name

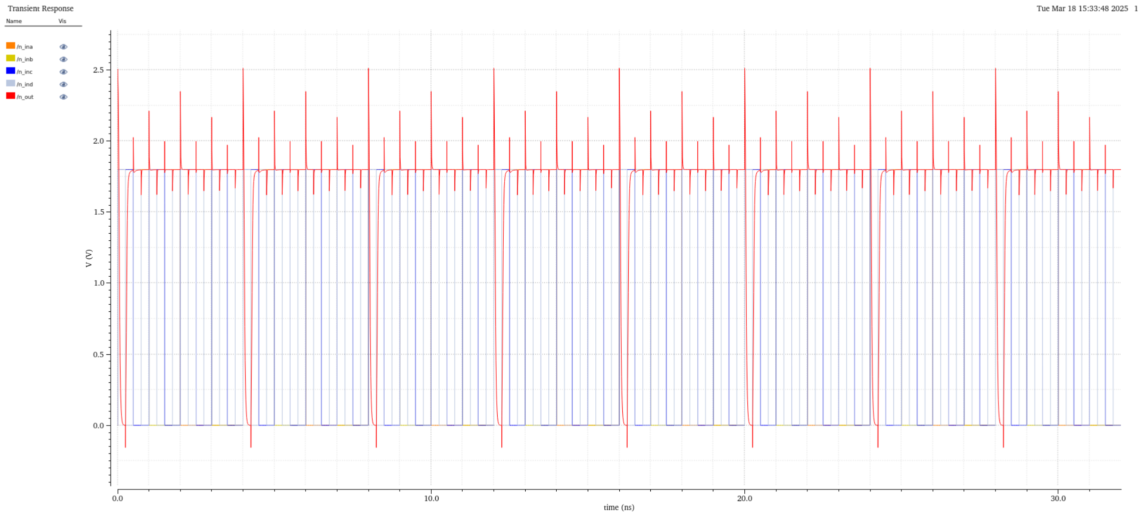
Click here

Button appears, click, schematic appears

The signal to be plotted can be selected in the schematic.

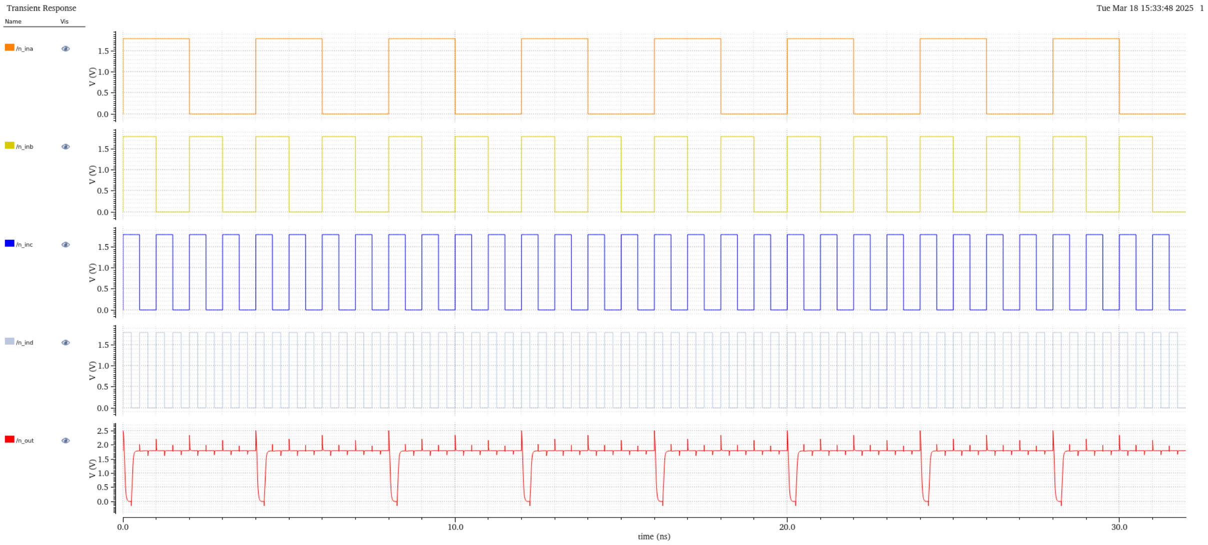
IV.2.3 Implementation of the simulation, representation of the signals

When all desired signals are assigned:
Main menu: Simulation/Netlist and run
Signal window opens:



Display signals individually below each other:

Main menu of the plot window: Graph / Split Current strip / Trace



Save waveform window to file or clipboard: Enter in log file File/save Image

Time domain measurements: Measurements / Transient
Rise times, fall times, delays available
Right-click on the signal name: Menu appears: Measurement / Time Domain / Delay
Use the plot to prove that the 4-fold NAND is working correctly. Plot in the log! When finished, close all simulation windows.

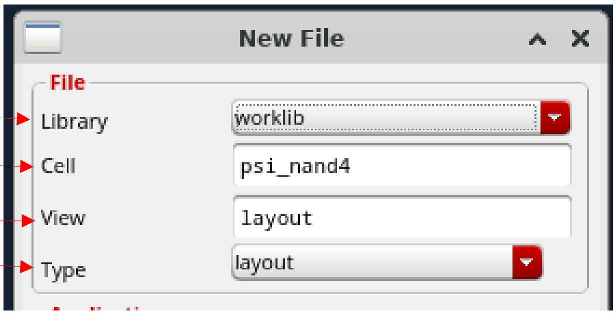
IV.3 Layout

In the Virtuoso Command Window: File/New/Cellview:

Working Library

Name (like Schematic)

Type Layout


New File

File

Library: worklib
Cell: psi_nand4
View: layout
Type: layout

Schematic window and empty layout window open:

Create layout with existing FETs:

Button at the bottom left or: Connectivity/Generate/All from Source, ok

Layout with FETs appears:

Place the FETs conveniently and wire them with rectangles and wires. Use the self-designed power rails for the power supply, perhaps even several next to each other!

In the supplied standard cell library, the distance between the li1 layers of the upper and lower power rails is 4µm, which should be maintained.
The busbars must be within the cell boundaries!
Using the /Edit/Flip command, transistors can be flipped so that drain/source regions at the same potential are directly opposite each other. Furthermore, FETs with gates that are to be connected should be placed so that the gates are exactly on top of each other. This allows them to be easily connected using the "wire" function.

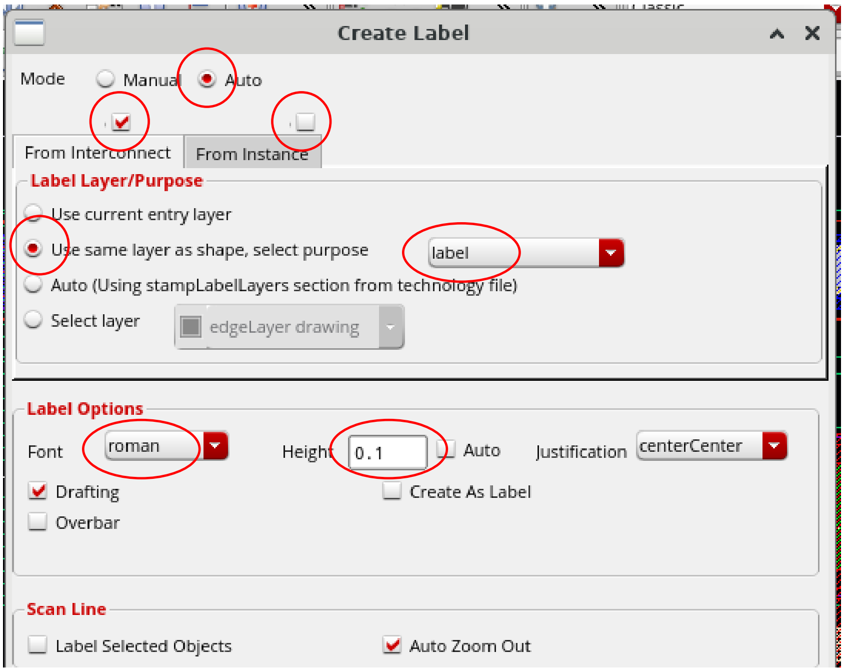
External connections:
The circuit's inputs and outputs require a "pin designator," which is a rectangle at the "Metal1 pin" level that is placed on the connection surface. Additionally, a "Metal1 drw" spot is required at the top hierarchy level; only this spot can be labeled.

Create label:

Assumption: Connection is on Metal1
Required: Pin designator = level Metal1 pin, additional spot placed on met1 drw

Label:
only met1 drw enable
shortcut**l**,F3
Adjacent window appears:

Fill out as shown:



Create Label

Mode ○ Manual ● Auto

☑ ☐

From Interconnect | From Instance

Label Layer/Purpose

○ Use current entry layer
● Use same layer as shape, select purpose: label
○ Auto (Using stampLabelLayers section from technology file)
○ Select layer: edgeLayer drawing

Label Options

Font: roman
Height: 0.1 ☐ Auto
Justification: centerCenter
☑ Drafting
☐ Create As Label
☐ Overbar

Scan Line

☐ Label Selected Objects
☑ Auto Zoom Out

When everything is done: Perform DRC, correct any errors

Further check: Connectivity/Check against source but returns cryptic error messages

The next points are still missing in the script:

Next Steps:
Pegasus DRC
Pegasus LVS (Layout versus schematic)
Pegasus Parasitics Extraction
Spectre Timing Simulation