Digital ICs Lab 2

Layout of an AND Gate

Contents

List of Figures

1 Preparatory Exercises

The exercise takes about 2 to 3 hours. Start by reading through this compendium. Use the information found here and the course book to fill in the answers to the preparatory exercises. The preparatory exercises MUST be completed before you arrive to the lab exercise. If you fail to do so you will not be allowed to participate.

1.1 Exercise 1

In Fig. 1 some objects have been marked. A cross section of an inverter implemented in a *gpdk*45 N-well CMOS process is shown below. Even though our process have 11 available metal layers only the two closest to the substrate are shown. Further, the CMOS process we use in the lab is a dual-well (sometimes called twin-well) process. However, the P-well is automatically generated for all areas not covered with N-well.

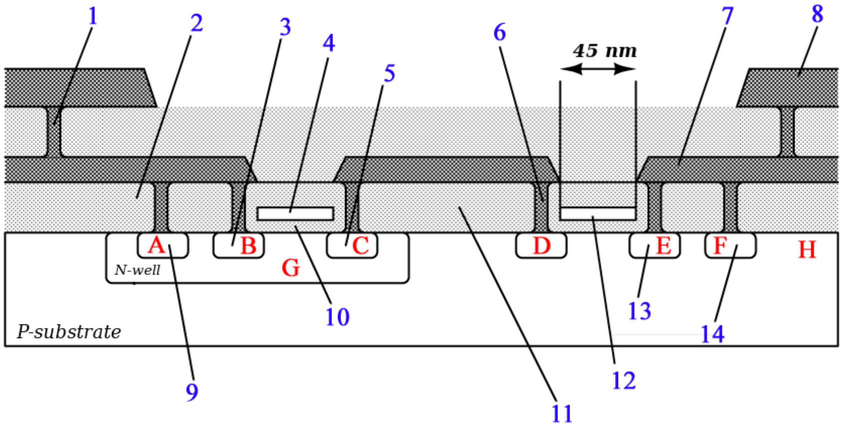


Figure 1: Cross section of an inverter

- Below is a description of the objects numbered 1 – 14 in Fig. 1. Assign the numbers to the description that best describe each object.

Field oxide: —— and ——

Gate oxide: ——

P-well (substrate) contact: ——

N-well contact: ——

PMOS source: ——

NMOS gate: ——

VIA : ——

Metal 1: ——

Metal 2: ——

PMOS drain: ——

NMOS source: ——

PMOS gate: ——

NMOS drain contact: ——

- What is the doping level (p+, n+, p-, n-) at markers A to H in Fig. 1?
  - Here the plus sign indicates heavily doped and the minus sign indicates lightly doped.
  - Hint! Look at pages 44-45 in the course book.

A:

B:

C:

D:

E:

F:

G:

H:

- What is the purpose of substrate and well contacts?

- Draw the schematic of an AND gate (assume there are no inverted input signals available).

1.2 Exercise 2

- Use the list of available layers and contacts for the *gpdk*45 process at the back of this compendium to explain the markings in the Fig. 2.

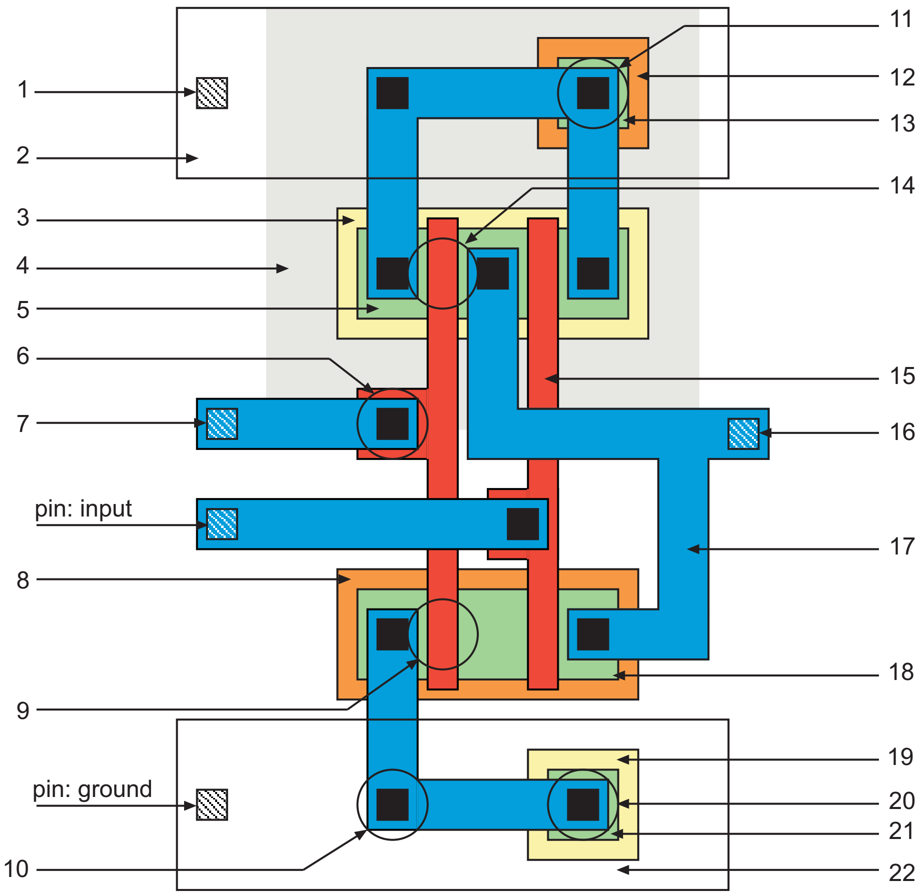
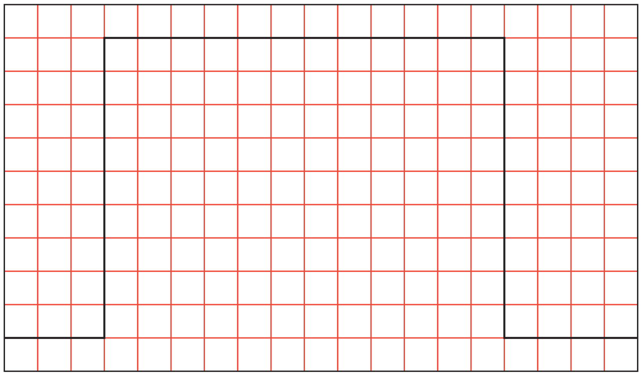


Figure 2: Layout of a logical gate

1. Pin:
2. Layer:
3. Layer:
4. Layer:
5. Layer:
6. Contact:
7. Pin:
8. Layer:
9. Transistor:
10. Contact:
11. Contact:
12. Layer:
13. Layer:
14. Transistor:
15. Layer:
16. Pin:
17. Layer:
18. Layer:
19. Layer:
20. Contact:
21. Layer:
22. Layer:

- What is the logical function of the gate in Fig. 2?

- The input signal to an inverter is shown below. Sketch the estimate of output signal and indicate the propagation delay t_{pHL} and t_{pLH}.



- What is the total propagation delay, t_p, for the inverter expressed in t_{pHL} and t_{pLH}?

Show your preparatory exercises to the lab assistant!

____

Date and signature of lab assistant

2 Circuit Layout of an AND gate

2.1 Purpose of this exercise

In this exercise you will create the circuit layout of an AND gate. You will perform circuit simulations on both the schematic and the layout level of your design. In order to complete the task in 4 hours it is highly recommended that you read through this lab compendium before attending the lab. Also, do not forget to complete the preparatory exercises before the lab. After completing this lab you will have basic knowledge on how to do full custom layout of an integrated circuit in a modern design tool.

2.2 The IC design tool

The software to be used is Cadence Virtuoso. This is a tool set commonly used in the industry for development of full custom integrated circuits. All designs in Cadence are organized in libraries. Reference libraries include basic building blocks such as transistors, resistors, capacitors etc. Design libraries include your designs in which you have instantiated components from the reference libraries. The Library Manager, which usually pops up when you start Cadence, shows a tree structure with the names of all the libraries available. You can add and/or remove libraries from the Library Manager. Each library includes cells and their different views. A cell is for example an inverter and the view, e.g., is a schematic, symbol or a layout representation of the cell. The simulation tool is called Analog Environment and is invoked from the menus in the schematic you want to simulate. All the supply sources and signal sources are inserted directly into the schematic. In this way we save a lot of work compared to writing all the netlists manually. Analog Environment uses the Cadence SPECTRE simulator, which is a SPICE-like simulator.

2.3 Initialization of the lab

To initiate the lab and setup the Linux environment, open a terminal window by clicking the right mouse button and select terminal. Start by logging in to the Cadence server *naum* via the intermediate server *ssh* by running the commands below in sequence

```
ssh -X ssh.edu.liu.se

ssh -X naum.ad.liu.se
```

Proceed by running the commands below one-by-one in your home directory.

Warning: Run the script *daisyCreateProj.sh* below only once the very first time you work with Digital ICs computer labs. Although low risk, you might loose data if you run it again when you have an existing project directory.

```
tcsh

source /sw/defaults/dotfiles/.tcshrc

module add kurs/eks

module add DIGIC
```

```
daisyCreateProj.sh  DIGIC

source  .DIGIC_rc

cad
```

The shell script *daisyCreateProj.sh* will create a directory in your home directory called DIGIC (this is also what we from now on refer to as your work area, or $WORKAREA = $HOME/DIGIC). Further on, it will inside that directory create some other directories as well as links to files and directories. For now, you do not have to worry too much about them.

Note: If you have to restart Cadence you should always start the tool in your home directory as follows:

```
tcsh

source  /sw/defaults/dotfiles/.tcshrc

module  add  kurs/eks

module  add  DIGIC

source  .DIGIC_rc

cad
```

2.4 Layout design of a NAND gate

2.4.1 Getting started

In this section you will learn how to use Cadence layout editor (Virtuoso) and handle its most elementary functions. Go to the window called Library Manager. Click at the library called “layoutExercise” with the left mouse button (lmb). You should now be able to see a list of cells in this library. To make your work easier we have prepared some cells for you. The hierarchic view for the AND gate is shown in Fig. 3. Both the inverter and the NAND gate are created separately and then connected, to form the AND gate. To test the functionality of the AND gate the top level of the design contains a testbench. The schematic and layout of the inverter have already been completed. You may open the designs by first selecting the cell to be opened with the lmb. Now, open the view by selecting it and double click lmb. The view named “layout” contains a layout, and “schematic” contains a schematic description of the circuit.

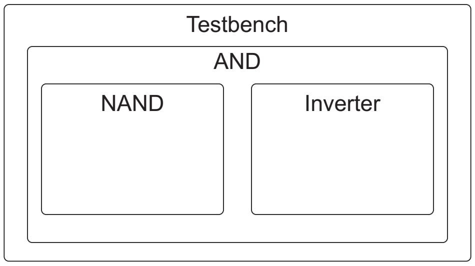


Figure 3: Hierarchical decomposition of the AND gate

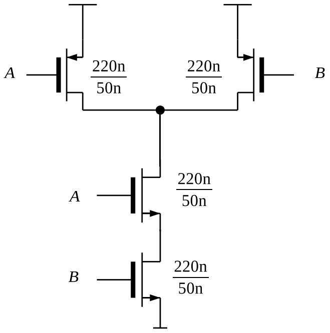


Figure 4: Transistor sizes in the NAND gate

For the NAND gate, only the schematic is provided. Your first task will be to complete the layout of this circuit. In order to ensure that the NAND gate and inverter can be interconnected the power supply wires have already been placed, you should not move these wires. When completing the layout it is recommended that you use the same layout style as shown in Fig. 2. Although several alternatives are possible this layout style fits well with the already prepared inverter circuit. The size of the transistors in the NAND gate are shown in Fig. 4. When designing the circuit, look at the design rules and use the measurement tool (Sec. 2.4.7) to minimize the number of design errors. You can also run the DRC tool (Sec. 2.5) regularly while designing the circuit. However, an incomplete layout may contain many errors that will eventually be eliminated when the design is finished. A summary of available layers and contacts in the process are given in the back of this compendium along with a list of some useful short cut keys and a list of some common design rules. In the text below reference to these keys are used, all functions are, however, also accessible throughout the menus.

Start by opening the layout of the NAND gate. Open the layout view of the NAND cell in the “layoutExercise” library.

2.4.2 Adding an instance

The first thing to do is to place the transistors, press i in the Virtuoso Layout Editing window to bring up the “create instance” pop-up menu. Press the browse button to open the instance selection dialog box. Select the library “*gpdk*45” and the “layout” view of the NMOS transistor “nmos1v_lvt”. Close the selection window and place the transistor in the layout by clicking lmb. The Esc key can be used to cancel an ongoing operation or exit a specific mode. Press shift-f to see the content of the instance. Repeat the procedure for the other three transistors, the PMOS transistor, “pmos1v_lvt”, is found in the same library as the NMOS transistor.

2.4.3 Changing the properties of an instance

You will have to adjust the width of each transistor, to do so select the transistor (use lmb) and press q to bring up the object property window. Here you may change the properties of the object, for the transistor you may change e.g. the width and length of the device. Enter the appropriate width for the transistors, do not forget to add the correct suffix (n for nano and without any space, e.g. 45*n*) since the values are given in meters. Other available options may be used to change the appearance of the device, e.g. contacts, guard rings and substrate contacts may be added or removed. Try selecting some of the options and click apply to see the effect on the device.

2.4.4 Moving the instance

After adjusting the transistors you have to move them into place. Select one transistor at a time and move it by pressing m. The transistors may be rotated, flipped and so on by clicking the right mouse button (rmb) during the move procedure. Please note that the status bar at the bottom of the Virtuoso window contain information on the function of the mouse buttons. During the move operation clicking the rmb will for example rotate the object. To flip or mirror the object you can use F3. If two identical objects are placed on top of each other, e.g. two contacts, they will be regarded as a single object. To minimize the required area you may want to place the source (or drain) of the PMOS transistors on top of each other since they are connected to the same node. By default the option "gravity" is enabled, this may cause problems when moving the objects, to disable this function press shift-e and deactivate the "gravity" option.

2.4.5 Adding wires

The next thing to do is to add the wires. On the left side of the layout window, you can find a subwindow called "layers". This subwindow shows all available layers. To draw a wire, first select the drawing layer, then press p, click and release lmb to start drawing the wire. To stop drawing (end of path), press enter or alternatively double click. A list of the available layers are presented at the back of this compendium. Two objects are connected if they use the same layer and part of them overlap. After drawing the wire, select it and press q to edit its properties. In the property window you may enter the width of the wire, change the layer and so on. The wire may be moved by selecting it and then pressing m. You may also move the corner points of a wire by placing the pointer over it and click lmb. If you have problems selecting the point of the wire you may try to press F4. Even though the process supports four metal layers you should only use metal layer one and two for wires in this exercise.

2.4.6 Adding contacts (VIAs)

You can add contacts by pressing "o" and selecting the contact. The placement is done in the same way as for an instance. The available contacts are listed at the back of the compendium.

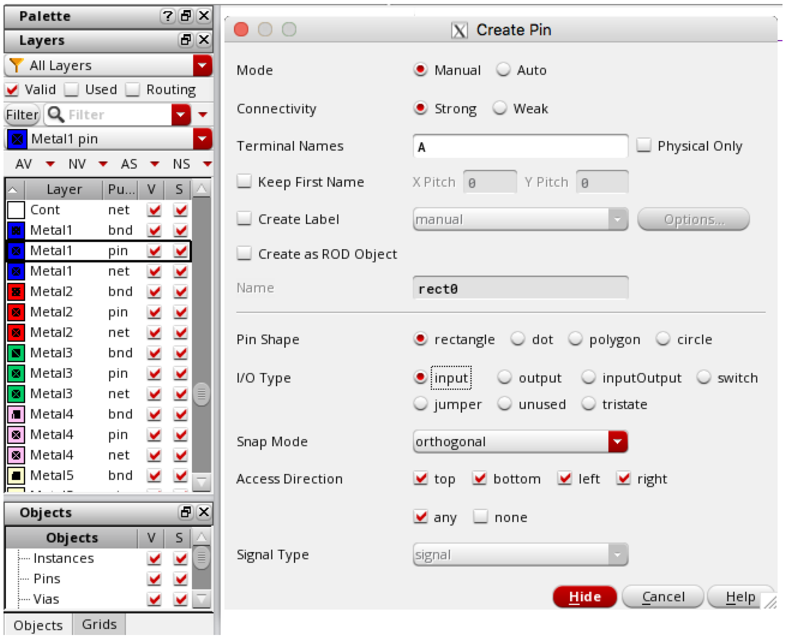
2.4.7 Distance measurements

To measure the distance between objects in the layout a measurement tool is available. Press "k" and then the lmb to start the measurement. Use shift-k to remove all rulers.

2.4.8 Adding Pins

"Pins" are used to mark the terminals of the gate. A terminal is a connection point to which external signals are connected. In this case we have five terminals, the two inputs, the output and the positive and negative power supply.

To create input and output pins, select Metal 1 – pn (its different from Metal 1 – drawing) in the layer subwindow. Press ctrl-p to bring the "create pin" window, if available select the mode "shape pin", the view shown in Fig. 5 appears. Specify Terminal Names and also specify the I/O type (In our example here all pins are defined as Input/Output in the schematic) and press "Hide". Now move the mouse back to the layout window and left click and release to place the first corner of pin A, where you want to place the terminals (see Fig. 1.2), click lmb again to select the second corner. Be sure to fit the complete pin within the area that it connects to, i.e., the pin should not be larger than the wire it is placed on. Continue to place the remaining pins. Create the positive and negative power supply pins in the same way, but in different layer (Metal 2). The pins you will create MUST be named in the same way as they are in the schematic view of the gate. However, we should use global definition of power supply pins i.e. “vdd!” and “gnd!” for naming the these pins.



☑ Valid ☐ Used ☐ Routing

Metal1 pin

Layer	Pu...	V	S
Cont	net	☑	☑
Metal1	bnd	☑	☑
Metal1	pin	☑	☑
Metal1	net	☑	☑
Metal2	bnd	☑	☑
Metal2	pin	☑	☑
Metal2	net	☑	☑
Metal3	bnd	☑	☑
Metal3	pin	☑	☑
Metal3	net	☑	☑
Metal4	bnd	☑	☑
Metal4	pin	☑	☑
Metal4	net	☑	☑
Metal5	bnd	☑	☑

Objects	V	S
Instances	☑	☑
Pins	☑	☑
Vias	☑	☑

Mode ● Manual ○ Auto

Connectivity ● Strong ○ Weak

Terminal Names A ☐ Physical Only

☐ Keep First Name X Pitch 0 Y Pitch 0

☐ Create Label manual Options...

☐ Create as ROD Object

Name rect0

Pin Shape ● rectangle ○ dot ○ polygon ○ circle

I/O Type ● input ○ output ○ inputOutput ○ switch ○ jumper ○ unused ○ tristate

Snap Mode orthogonal

Access Direction ☑ top ☑ bottom ☑ left ☑ right ☑ any ☐ none

Signal Type signal

Figure 5: Adding pins window

2.5 Running the design rule check (DRC) tool

To check that no design rules are violated you must use the design rule check (DRC) tool. During the DRC all design rules, e.g., minimum spacing, minimum dimensions and so on, for the process are checked.

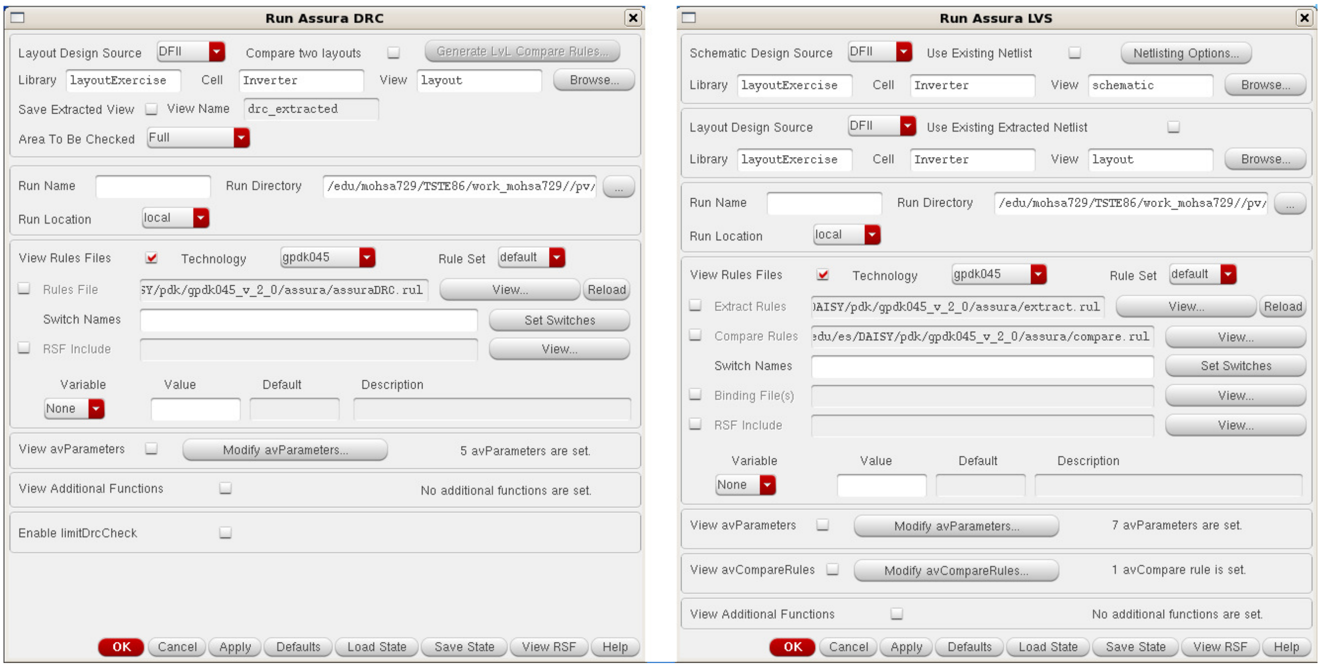
To start the DRC, select “Assura ⇒ Run DRC...” in the menu. Before clicking the apply button in the DRC window (Fig. 6), set the “Technology” option to “gpdk045” and ensure that it uses the correct design rule file as following:

$$\$PDKAREA/assura/assuraDRC.rul$$

Run the DRC by clicking ok. A few message windows might appear asking if you wish to save the data, overwrite data or that another job is active, if so click Yes on all of these. Once the run is completed a message box will appear asking if you wish to view the results of the run, click Yes. All violations are indicated by a flashing marker in the Virtuoso window. If there were any errors they will also appear in the Error Layer Window (ELW), if so correct them before you proceed. Along with the error a short explanation is given, to display the reason for a specific error in the layout, click on (left/right) arrows in the ELW window. If you get the “nwell_conn_StampErrorFloat” error you must check so that you have connected the power supply correctly. “nwell_conn_StampErrorFloat” informs you about the fact that the well contact is either not placed within the N-well, or the N-well is not connected to the power supply (vdd!). It can be because your vdd! pin is not properly created. To get over this error look more carefully on the Fig. 6. Read the error messages and think about how to fix them! You can also get help by looking at the documents directory of the *gpdk*45 here:

$\$PDKAREA/docs/$

and specially the *gpdk*45 DRC document.



Run Assura DRC

Layout Design Source: DFII — Compare two layouts ☐
Library: layoutExercise — Cell: Inverter — View: layout
Save Extracted View ☐ — View Name: drc_extracted
Area To Be Checked: Full
Run Name: — Run Directory: /edu/mohsa729/TSTE86/work_mohsa729//pv/
Run Location: local
View Rules Files ☑ — Technology: gpdk045 — Rule Set: default
☐ Rules File: 3Y/pdk/gpdk045_v_2_0/assura/assuraDRC.rul
Switch Names:
☐ RSF Include:
Variable: None — Value — Default — Description
View avParameters ☐ — 5 avParameters are set.
View Additional Functions ☐ — No additional functions are set.
Enable limitDrcCheck ☐

Run Assura LVS

Schematic Design Source: DFII — Use Existing Netlist ☐
Library: layoutExercise — Cell: Inverter — View: schematic
Layout Design Source: DFII — Use Existing Extracted Netlist ☐
Library: layoutExercise — Cell: Inverter — View: layout
Run Name: — Run Directory: /edu/mohsa729/TSTE86/work_mohsa729//pv/
Run Location: local
View Rules Files ☑ — Technology: gpdk045 — Rule Set: default
☐ Extract Rules:)AISY/pdk/gpdk045_v_2_0/assura/extract.rul
☐ Compare Rules: edu/es/DAISY/pdk/gpdk045_v_2_0/assura/compare.rul
Switch Names:
☐ Binding File(s):
☐ RSF Include:
Variable: None — Value — Default — Description
View avParameters ☐ — 7 avParameters are set.
View avCompareRules ☐ — 1 avCompare rule is set.
View Additional Functions ☐ — No additional functions are set.

Figure 6: Example of ASSURA DRC and LVS tools’ menus

2.6 Running the layout versus schematic (LVS) tool

In order to assure that the layout description of the circuit agrees with the schematic the layout vs. schematic (LVS) tool is used. The tool builds a circuit netlist from the layout view and compares it with the netlist of the schematic view. Any deviations are reported in a log-file. In the layout window, select “Assura ⇒ Run LVS...”. The LVS window appears. Before running the tool you must verify that the correct views are used as inputs, i.e., schematic and layout of the NAND gate. Your LVS window should now look similar to Fig. 6.

Press OK to start the LVS. Wait for some seconds that a pop-up window appears, indicating whether the LVS was successful or not. If the LVS was unsuccessful you need to check the icfb window for error messages. If the LVS was successfully carried out this only means that the test was performed, not that your circuit is error free. The log-file contains information about the run. If the circuit do not have any errors the text “The net-lists match.” will be displayed after the initial circuit summary. However if an error is found the latter part of the log-file contains additional information. Press the “OK” button to see the LVS debug window. The “LVS Debug” window can be used to figure out which components do not match in layout and schematic. Play around with this debug tool to identify the errors and go back to the layout and correct them. Unfortunately some of the error messages might be hard to interpret, however, when an error occurs it is a good idea to go through all wiring and check so that all the connections are correctly done. Further, check the size of all devices, are the width and length of the transistors same in layout and schematic? Are the pins named correctly? Have you placed all pins in the layout? A tricky error to detect might be if you switch the A and B inputs, as you see in Fig. 2.2 the source of the transistor connected to input B is connected to ground. If you switch the inputs this will no longer be true and an error will be generated. You may also use probe and zoom functions in debug window and run LVS until the netlists match

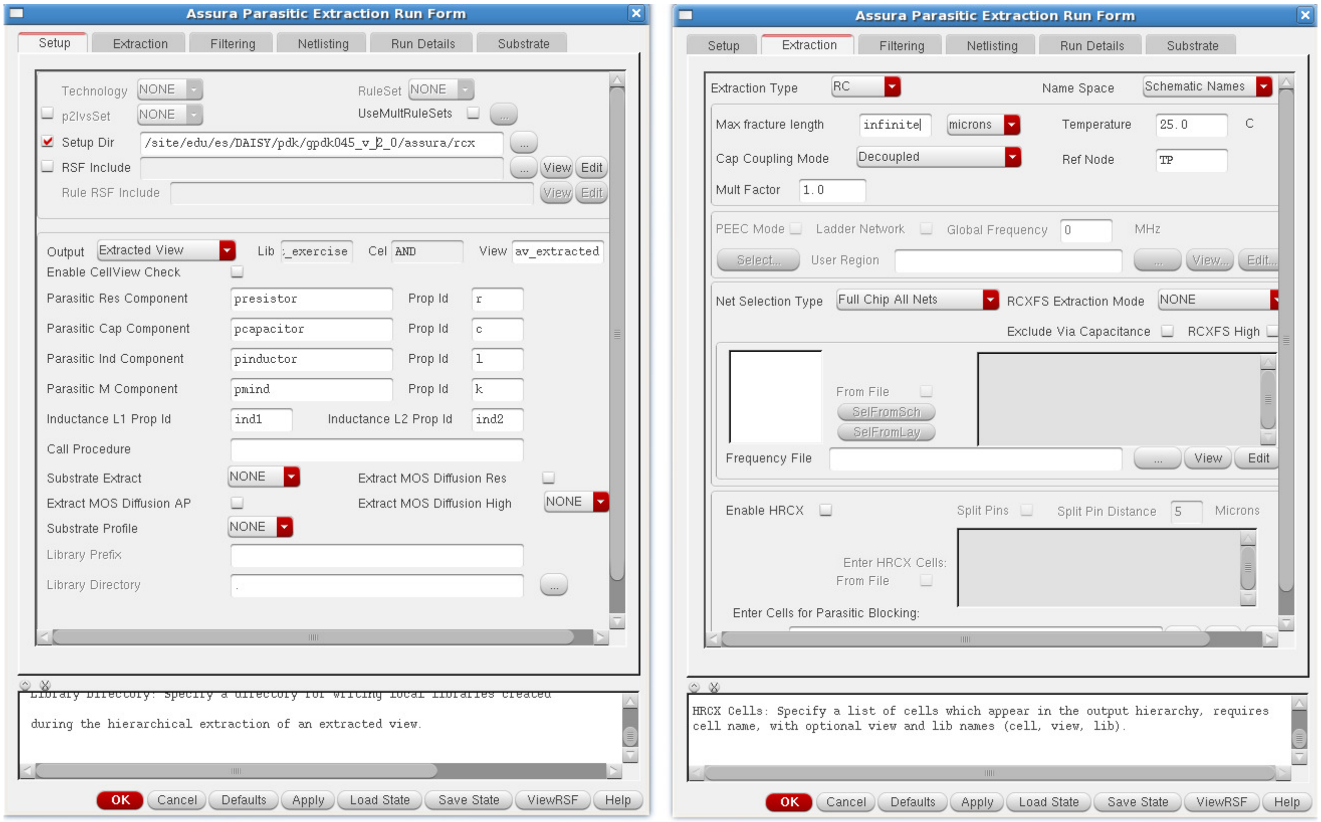
2.7 Assembling the AND gate

It is now time to assemble the AND gate, which is done be editing the AND gate *only* (do not edit the inverter or NAND gate). As said before only the schematic view of AND gate is created for you but not the layout view. To create layout view click on the cell “AND”, then go to the “File ⇒ New ⇒ Cell View ...” in the Library manager and select layout from the drop down menu of “type” and click OK. Now that you have the layout window of AND gate, you can instantiate NAND and Inverter gates. Here we will use the layout of the NAND gate that you have just finished and connect it to the inverter that has been prepared earlier. Start by placing one instance of the NAND gate and one of the inverter gate in the layout view “AND” by pressing i. Align the block so that the power supply wires are connected. Connect the output of the NAND gate to the input of the inverter. Also **add pins** to the input (“**A**” and “**B**”) and output (“**out**”) nodes of the AND. You will also need to place pins at the positive and negative power supply voltages, use the names “**vdd!**” and “**gnd!**” respectively. Do not forget to run the DRC tool on the complete design. Once you have finished the layout without any error, you should run LVS tool for the AND gate to ensure that it is the same as schematic.

2.8 Running the parasitic extraction tool

The parasitic extraction tool is used to extract all layout dependent parasitic such as capacitive coupling between wires. If simulation is carried out only on the schematic level of a circuit these parasitic are not taken into account, it is therefore important to do simulations on the circuit layout as well. The parasitic extraction tool is started by selecting “Assura ⇒ Run Quantus QRC...”. The window shown in Fig. 7 appears in which we can specify details on which type of extraction to perform.

Now look at the Extraction tab as shown in Fig. 7 right side. The standard setting is “C Only”. This extracts only the capacitances between the nets in the layout which might be enough in most cases for small cells. Furthermore, the “RC” extraction takes also parasitic resistors into account and is more accurate; however, it may take longer time to finish the extraction in huge designs. Here let’s try “RC” parasitic extraction. Continue by specifying “**gnd!**” as reference in the field “Ref Node”. Run the extraction tool by clicking “OK”. A new cell view, “av_extracted”, will appear in the library manager. This cell view is similar to the layout, however all parasitic that are extracted are included. Opening the view you will see all parasitic capacitors, resistors and diodes in the design. Now we are ready to simulate our designed AND gate.



Technology NONE
RuleSet NONE
□ p2lvsSet NONE
UseMultRuleSets □
☑ Setup Dir /site/edu/es/DAISY/pdk/gpdk045_v_2_0/assura/rcx
□ RSF Include
Rule RSF Include
Output Extracted View Lib _exercise Cel AND View av_extracted
Enable CellView Check □
Parasitic Res Component presistor Prop Id r
Parasitic Cap Component pcapacitor Prop Id c
Parasitic Ind Component pinductor Prop Id l
Parasitic M Component pmind Prop Id k
Inductance L1 Prop Id ind1 Inductance L2 Prop Id ind2
Call Procedure
Substrate Extract NONE
Extract MOS Diffusion Res □
Extract MOS Diffusion AP □
Extract MOS Diffusion High NONE
Substrate Profile NONE
Library Prefix
Library Directory .

Library Directory: Specify a directory for writing local libraries created during the hierarchical extraction of an extracted view.

Extraction Type RC
Name Space Schematic Names
Max fracture length infinite microns
Temperature 25.0 C
Cap Coupling Mode Decoupled
Ref Node TP
Mult Factor 1.0
PEEC Mode □ Ladder Network □ Global Frequency 0 MHz
User Region
Net Selection Type Full Chip All Nets
RCXFS Extraction Mode NONE
Exclude Via Capacitance □ RCXFS High □
From File □
Frequency File
Enable HRCX □
Split Pins □ Split Pin Distance 5 Microns
Enter HRCX Cells:
From File □
Enter Cells for Parasitic Blocking:

HRCX Cells: Specify a list of cells which appear in the output hierarchy, requires cell name, with optional view and lib names (cell, view, lib).

Figure 7: Parasitic extraction tool

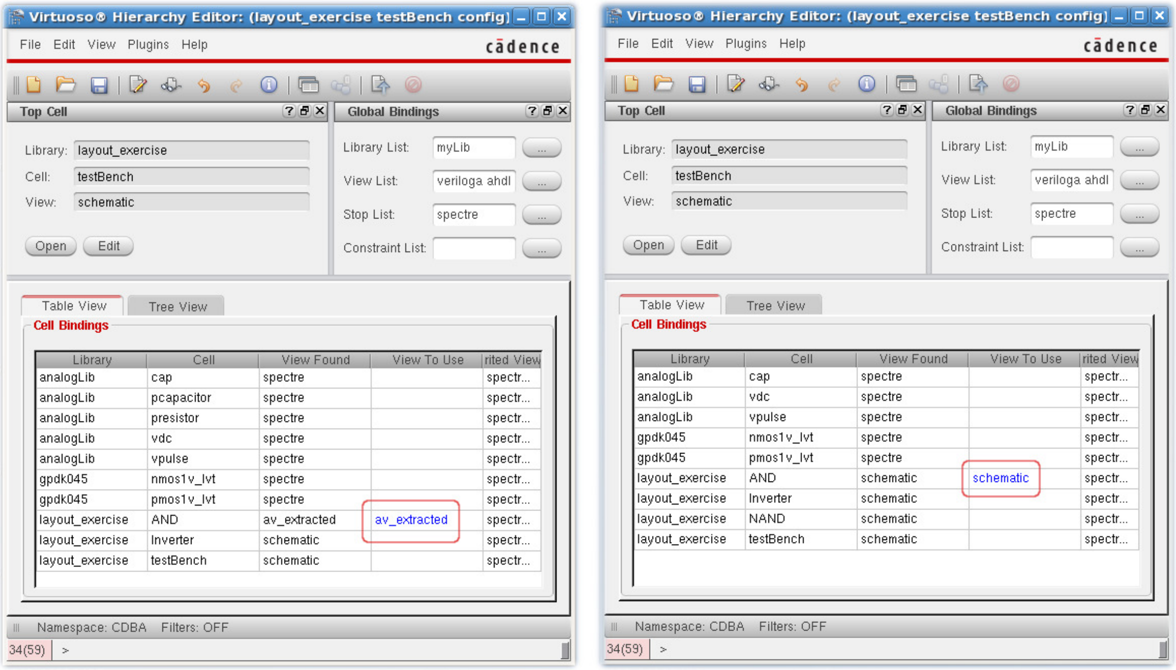
2.9 Simulating your design

It is now time to simulate your design and compare its performance with that of the schematic. In the library manager, select the “TestBench” and open the view “config”. In the dialog box, select “yes” on both questions. Two windows will open, the schematic view of the circuit and the hierarchy editor, Fig. 8. The hierarchy editor is a configuration tool for the designs where you, for example, may select the model to be used for a specific part of your design. For example some parts may be implemented using VHDL or Verilog in which the circuit is described using a high-level programming language while other parts are pure transistor level implementations. In our case we have two descriptions for the AND gate, one schematic view and an extracted view. We will start by simulating the schematic description of the circuit.

2.9.1 Schematic simulation

In the hierarchy editor right click in the “view to use” field of the AND gate (marked in the figure) and select schematic. Update the design by selecting “View ⇒ Update”, click “ok” in the pop-up window. Save the settings by selecting “File ⇒ Save”. In the schematic window, start the circuit simulator by selecting “Launch ⇒ ADE L”. The analog design environment, Fig. 9, will be started.

First you should set which simulator to use by selecting “Setup ⇒ Simulator/Directory/Host...’. A dialog is shown where you select “spectre’ as simulator at the top and then press “OK”. Continue by loading some predefined design variables from testbench. For that, select “Variables ⇒ Copy from Cellview”. Set the load capacitor to 1f Farad (you just need to type 1f in the field).


Virtuoso® Hierarchy Editor: (layout_exercise testBench config)
File Edit View Plugins Help
cādence
Top Cell
Global Bindings
Open
Edit
Table View
Tree View
Cell Bindings
Namespace: CDBA Filters: OFF
34(59)

Top Cell — Library: layout_exercise; Cell: testBench; View: schematic

Global Bindings — Library List: myLib; View List: veriloga ahdl; Stop List: spectre; Constraint List:

Left window:

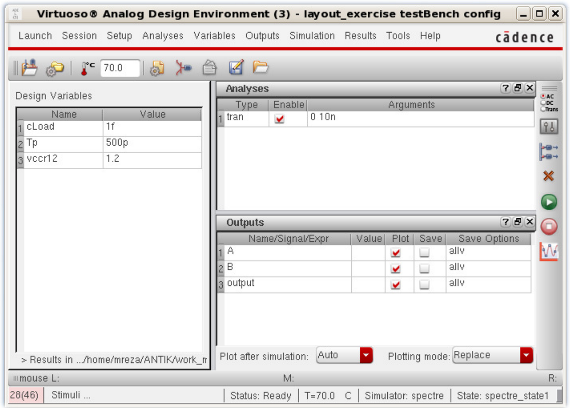
Library	Cell	View Found	View To Use	rited View
analogLib	cap	spectre		spectr...
analogLib	pcapacitor	spectre		spectr...
analogLib	presistor	spectre		spectr...
analogLib	vdc	spectre		spectr...
analogLib	vpulse	spectre		spectr...
gpdk045	nmos1v_lvt	spectre		spectr...
gpdk045	pmos1v_lvt	spectre		spectr...
layout_exercise	AND	av_extracted	av_extracted	spectr...
layout_exercise	Inverter	schematic		spectr...
layout_exercise	testBench	schematic		spectr...

Right window:

Library	Cell	View Found	View To Use	rited View
analogLib	cap	spectre		spectr...
analogLib	vdc	spectre		spectr...
analogLib	vpulse	spectre		spectr...
gpdk045	nmos1v_lvt	spectre		spectr...
gpdk045	pmos1v_lvt	spectre		spectr...
layout_exercise	AND	schematic	schematic	spectr...
layout_exercise	Inverter	schematic		spectr...
layout_exercise	NAND	schematic		spectr...
layout_exercise	testBench	schematic		spectr...

Figure 8: Hierarchy editor window

The input signals A and B, are periodic pulses with a period of Tp second. Set the “Tp” to 500p second and use power supply of 1.2 volts for “vccr12”.


Virtuoso® Analog Design Environment (3) - layout_exercise testBench config
Launch Session Setup Analyses Variables Outputs Simulation Results Tools Help
cādence
70.0
Design Variables
Analyses
Outputs
mouse L:
M:
R:
28(46)
Stimuli ...

Design Variables:

	Name	Value
1	cLoad	1f
2	Tp	500p
3	vccr12	1.2

> Results in .../home/mreza/ANTIK/work_

Analyses:

	Type	Enable	Arguments
1	tran	✓	0 10n

Outputs:

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	A		✓	□	allv
2	B		✓	□	allv
3	output		✓	□	allv

Plot after simulation: Auto   Plotting mode: Replace

Status: Ready | T=70.0 C | Simulator: spectre | State: spectre_state1

Figure 9: Analog Design Environment window

Secondly you must add the signals to plot by selecting “Outputs ⇒ To be plotted ⇒ Graphics select on schematic” (in the ADE window) and click on the signal wires A, B in the schematic view to plot the voltage (do not select the input pin, which would plot the current through the pin). Do also select the output and press “esc”. Next, the type of simulation to run is chosen by selecting “Analysis ⇒ Choose”. In the pop-up window, select the transient (tran) analysis, a suitable stop time is 20n second (do not forget to use the suffix n, unless you want to stay all night!).

Start the simulation by selecting “Simulation ⇒ Netlist and Run” or alternatively by the green play button. To interrupt an ongoing simulation, use “Simulation ⇒ Stop”. When the simulation is completed (should take only a few seconds) the result will be displayed graphically in the“waveform window”. You may zoom on the curves by drawing a rectangular window using rmb of the mouse and to have a full view of curves press “f”. Start by verifying the functionality of the AND gate. You may have to zoom in closely on the curves and also split the waveforms from each other to clearly see the output signal. In order to measure the performance of the circuit we will have a look at the propagation delay through the AND gate. To measure the delay in the waveform window, use the markers available in the menu: Markers or alternatively just press “a” and click on the waveforms. Put the first marker on the first signal and the second marker on the second signal and read the time difference (delta) at the bottom of the waveform window.

- Calculate:

$$t_{pHL} = \rule{2cm}{0.4pt}$$

$$t_{pLH} = \rule{2cm}{0.4pt}$$

$$t_p = \rule{2cm}{0.4pt}$$

- Do the high-to-low (t_{pHL}) and low-to-high (t_{pLH}) propagation delays differ? Why?

2.9.2 Post-layout simulation

Now we want to simulate our designed circuits layout taking all extracted parasitic into account. Go back to the Hierarchy editor and change the “view to use” to “av_extracted” that means simulator will use the layout extracted view of the AND circuit, do not forget to update and save. Now run the simulation again. If simulation completes successfully you will have the result of the post layout simulation. Measure the propagation delay with the markers in the waveform window as before.

- Calculate:

$$t_{pHL} = \rule{2cm}{0.4pt}$$

$$t_{pLH} = \rule{2cm}{0.4pt}$$

$$t_p = \rule{2cm}{0.4pt}$$

- Compare and comment on the result of both simulations, why is the propagation delay different in the two cases?

- Do the result agree with what you expected? Explain why or why not.

- If possible, compare your result with that obtained by other lab groups. Try to explain any differences.

Show your layout and simulation results to the assistant!

Date & LiU-ID of Student	Signature of Lab Assistant

3 Appendices

3.1 Layers in *gpdk*45 process

Layer description	**Layer name**
Metal 4	Metal4
Metal 3	Metal3
Metal 2	Metal2
Metal 1	Metal1
Poly	Poly
N-well	Nwell
Negatively doped substrate	Nimp
Positively doped substrate	Pimp

3.2 Contacts in *gpdk*45 process

The contact creates a connection between two layers. A connection between two *metal* layers are called a VIA while a connection between other layers like Metal 1 and the N-well is called a cut.

Layer description	**Layer 1**	**Layer 2**
M2_M1	Metal2	Metal1
M1_PO	Metal1	Poly
M1_NWELL	Metal1	N-well
M1_PSUB	Metal1	P-sub

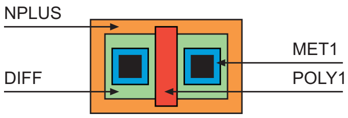
3.3 Some useful shortcut keys in Cadence Virtuoso©

c	copy, select the object to copy and press c.
m	move, select the object to copy and press m, press F3 will bring up other options such as flip, rotation and mirror.
p	create path, draw a wire.
s	changing the edges, extending area, path etc.
o	create contact, brings up the contact creation window.
r	create rectangle.
ctrl - p	create pin, brings up the contact creation window.
k	measurement tool, brings up a ruler for distance measurements. Click lmb to0select the starting point then point at the end point and click lmb again.
shift - k	remove all measurement rulers.
q	object properties, brings up the properties of the selected object.
shift -f	view the content of all instances.
ctrl - f	view only the content of the top-level.
delete (del)	Delete the selected objects.
escape (esc)	Cancel any ongoing operation, e.g. move.
i	place an instance.
f	zoom to fit
shift - z	zoom out
ctrl - z	zoom in
z	zoom (mouse selection)
F2	Save
F3	Flip, mirror, or rotate the object subject to a move or copy operation.

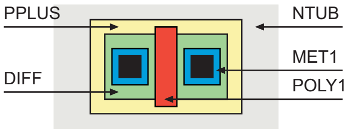
For list of daisy-specific bindkeys, use key F12

3.4 Transistors and substrate contacts in *gpdk*45 process

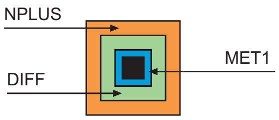
nmos4 transistor



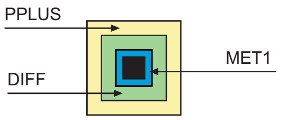
pmos4 transistor



ND_C contact



PD_C contact


PPLUS
DIFF
MET1