Tutorial:Layout Tutorial2

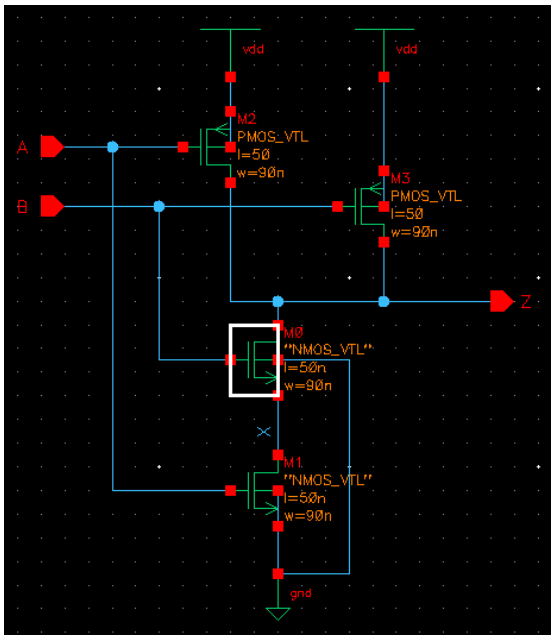
Layout Tutorial #2: Extraction and LVS

In this tutorial you will create the schematic and layout for a NAND gate, and then perform a layout-vs.-schematic (LVS) check to verify the connectivity. You'll also perform a parasitic extraction and generate an HSPICE netlist with accurate wire- and source-/drain, adjacent wires capacitances, as well as wire resistances generated from the layout.

Create the NAND2 Schematic

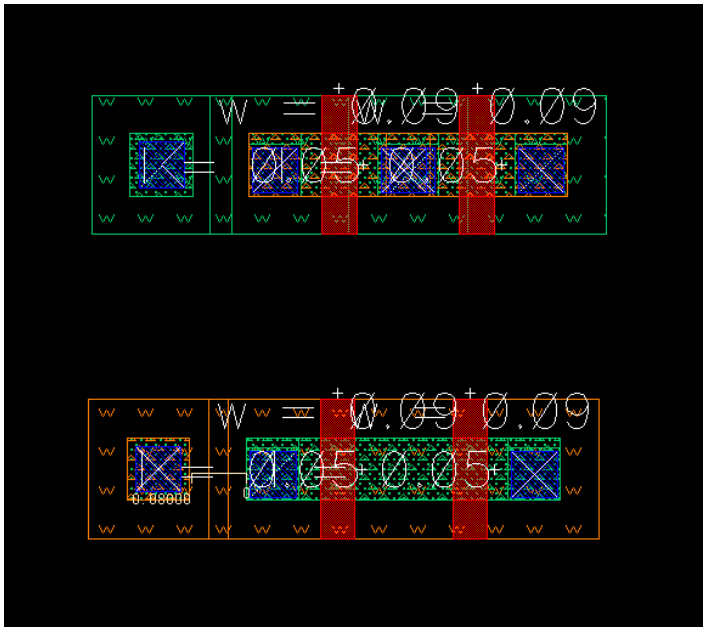
Create a cell called "nand2", and make a schematic like the one shown below. Note that the all transistors (NMOS and PMOS) have a width of 90 nm and length of 50nm. Note also that you can add a "Wire Name" to internal nodes by selecting **Create->Wire Name** or by hitting the "l" (lower case "L") key. The figure shows that the node between the two NMOS transistors has been given the name "X". When you generate an HSPICE netlist, this node will now have a more meaningful name, rather than something random like "NET41".

Check and save the schematic.

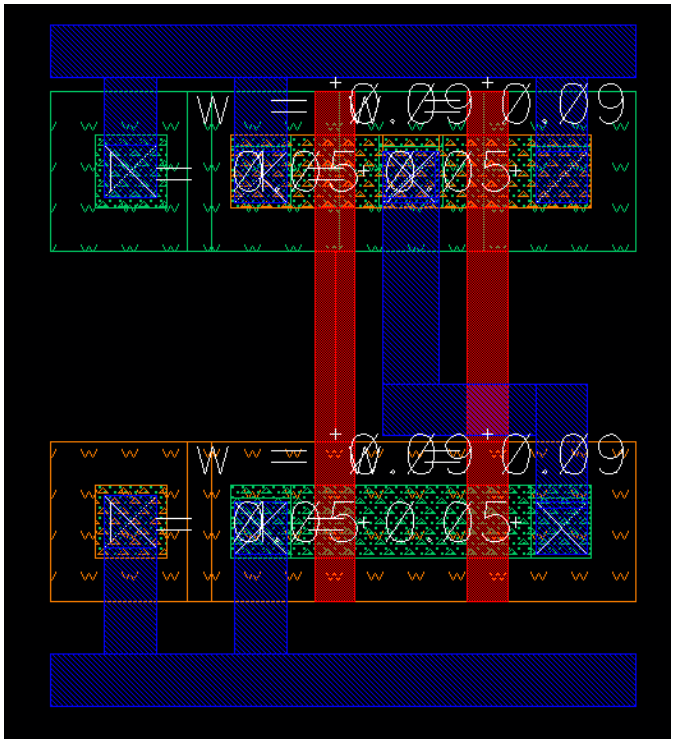


Create the NAND2 Layout

Now create the layout view. Create an instance of an NMOS transistor. Set Width to "90n M" and Length set to "50n M" and Fingers to 2. Also create two instances of PMOS transistors, with their Widths set to "90n M" and Length set to "50n M" .

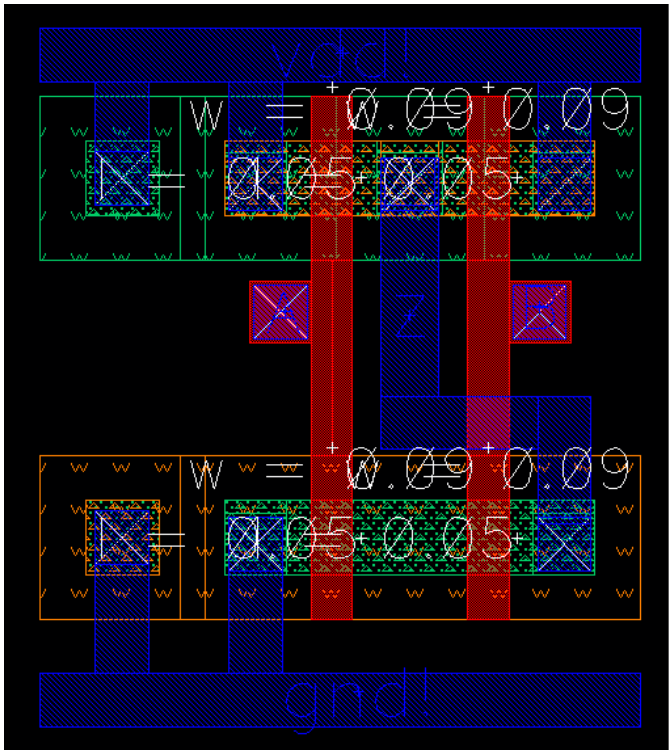


Now connect up the devices with metal1 and poly as shown below.



Next, we'll most likely want to connect the inputs to this cell using a metal wire to another cell. Create contacts to bring the poly wires up to metal1. Select **Create->Contact** or hit the "o"key. Set contact type to M1_POLY, and create contacts as shown below. Click "cancel" when done. Then create the remaining poly wire to connect the contact to the gates.

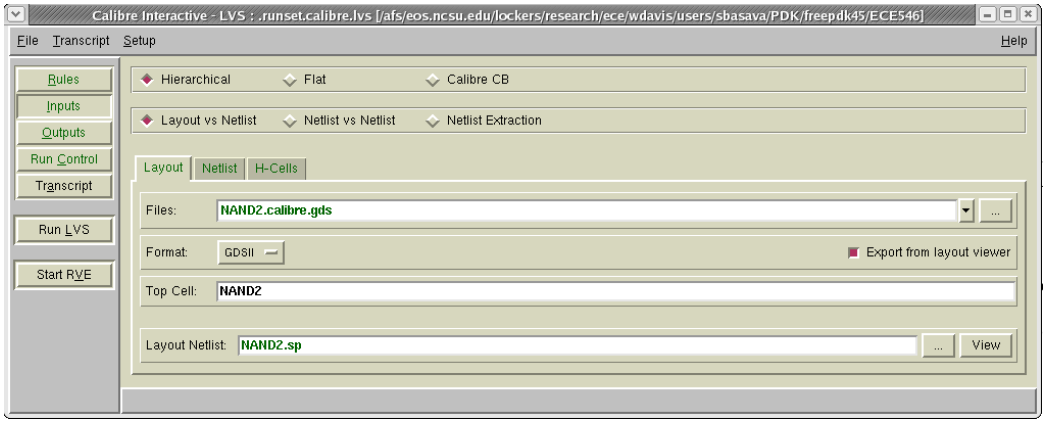
We need to add text labels for each of the pins on the corresponding layers, which will be used in LVS to recognize them as ports in the layouts. Select **Create->Label** or hit the "l" key. Enter label name like vdd!, gnd!, A and so on. Set height to 0.1 . Make sure that the layer on which you want to apply the label on, is currently being selected in LSW window.



When you are done creating the layout, save the design. Make sure that the layout is **DRC clean**.

LVS

To perform a layout-vs.-schematic (LVS), choose **Calibre->Run LVS...**. The LVS form appears, as shown below. If you do not see the window appear, or if you get an error, then it's possible that you didn't type "add calibre" as instructed above. You will need to exit Virtoso, log out, and log back in, setting up your environment in the correct order.



There are a number of options you need to set and know what they are.

Rules

Calibre-LVS Rules File and Calibre-LVS Run Directory are already filled in by the tool,leave them as it is.

Inputs

Select "Hierarchial", "Layout vs Netlist"

Under the **Layout** tab

Files : nand2.calibre.gds

Top Cell: nand2

Layout Netlist: nand2.sp

These options are already be filled in by the tool, leave them as is

Format: select "GDSII" and select the option "Export from layout viewer" (This is very important)

Under the **Netlist** tab

Files: nand2.src.net

Top Cell:nand2

These options are already be filled in by the tool, leave them as it is

Format: select "SPICE" and select the option "Export from schematic viewer" (This is very important)

Outputs

Under the **Report/SVDB**

LVS Report File: nand2.lvs.report

This option is already be filled in by the tool, leave it as is

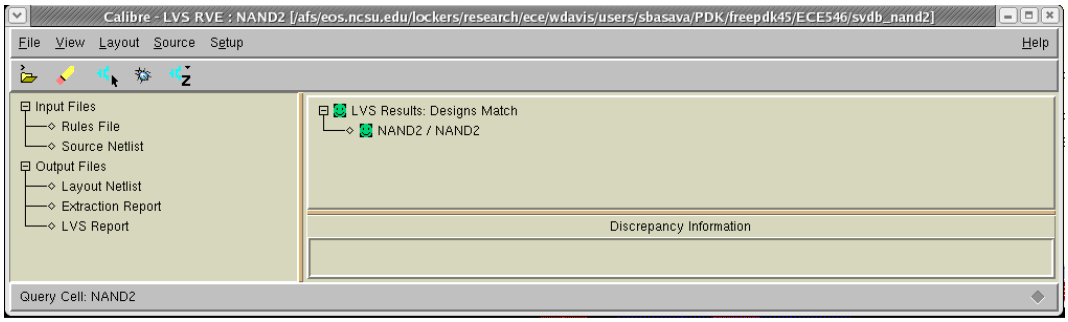
svdb directory: svdb_nand2

Select "View Report after LVS Finishes"

Perform an LVS Check without Errors

Set the LVS form with the options shown above. Then click the “Run LVS” button. If LVS runs sucessfully, with out any error, then you will see the below window with a smilie :)

Click on the "Transcript" tab in Calibre Interactive - LVS to see the log file.



The LVS Report File - NAND2.lvs.report is also opened and a part of the report is shown below

CELL COMPARISON RESULTS (TOP LEVEL)

```
          #         ####################          _    _
         #          #                  #         *    *
    #    #          #       CORRECT      #          |
     # #            #                   #         \___/
      #             ####################
```

LAYOUT CELL NAME: NAND2
SOURCE CELL NAME: NAND2

INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	5	5	
Nets:	6	6	
Instances:	2	2	MN (4 pins)
	2	2	MP (4 pins)
Total Inst:	4	4	

NUMBERS OF OBJECTS AFTER TRANSFORMATION

	Layout	Source	Component Type
Ports:	5	5	
Nets:	5	5	
Instances:	2	2	MP (4 pins)
	1	1	SMN2 (4 pins)

Total Inst: 3 3

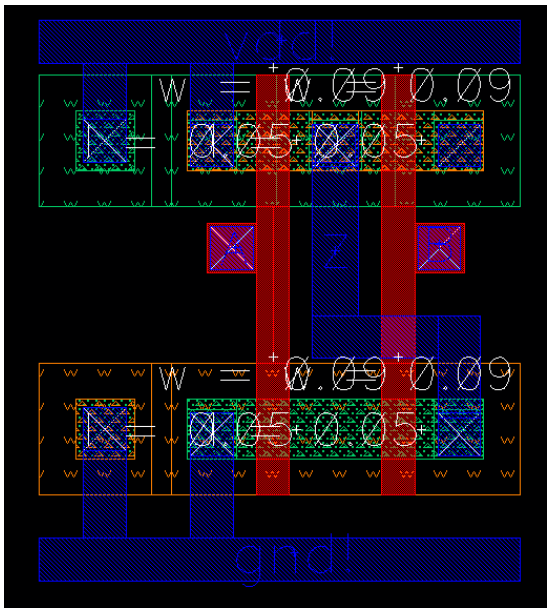
INFORMATION AND WARNINGS

	Matched Layout	Matched Source	Unmatched Layout	Unmatched Source	Component Type
Ports:	5	5	0	0	
Nets:	5	5	0	0	
Instances:	2	2	0	0	MP(PMOS_VTL)
	1	1	0	0	SMN2
Total Inst:	3	3	0	0	

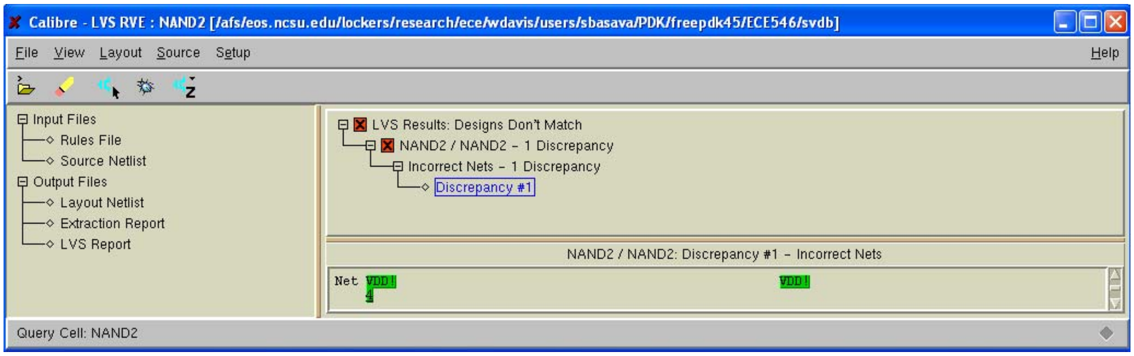
o Initial Correspondence Points:
Ports: GND! Z VDD! A B

Perform an LVS Check with Errors

Just as an example of what can go wrong when running LVS, try removing the piece of metal1 that connects the PMOS source node to VDD! in the upper right-hand corner of the layout, as shown below.



Then save the design and re-run the LVS check. You should see the following output.



and the Error report will be as follows

```
                 CELL COMPARISON RESULTS ( TOP LEVEL )

                  #     #          ####################
                   # #            #                    #
                    #             #     INCORRECT      #
                   # #            #                    #
                  #     #          ####################

  Error:      Different numbers of nets (see below).
  Error:      Connectivity errors.

LAYOUT CELL NAME:            NAND2
SOURCE CELL NAME:            NAND2

--------------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                   Layout      Source          Component Type
                   ------      ------          --------------
 Ports:               5           5
 Nets:                7           6      *

 Instances:           2           2             MN (4 pins)
                      2           2             MP (4 pins)
                   ------      ------
 Total Inst:          4           4

NUMBERS OF OBJECTS AFTER TRANSFORMATION
---------------------------------------

                   Layout      Source          Component Type
                   ------      ------          --------------
 Ports:               5           5
 Nets:                6           5      *
```

```
 Instances:              2              2              MP (4 pins)
                         1              1              SMN2 (4 pins)
                       ------         ------
 Total Inst:             3              3

          * = Number of objects in layout different from number in source.

                                  INCORRECT OBJECTS
LEGEND:
-------

   ne    = Naming Error (same layout name found in source
           circuit, but object was matched otherwise).

                                  INCORRECT NETS

DISC#    LAYOUT NAME                                                    SOURCE NAME

   1      Net VDD!                                                      VDD!
            4

                                  INFORMATION AND WARNINGS

                      Matched      Matched      Unmatched      Unmatched      Component
                       Layout       Source        Layout         Source        Type
                      -------      -------      ---------      ---------      ---------
    Ports:              5            5             0              0
    Nets:               6            5             0              0

    Instances:          2            2             0              0       MP(PMOS_VTL)
                        1            1             0              0       SMN2
                      -------      -------      ---------      ---------
    Total Inst:         3            3             0              0

o Initial Correspondence Points:
    Ports:          GND! VDD! Z A B
```

How would you figure out what the problem is from all this information? One thing that you should notice is that the number of nets is listed as 7 in the layout, but only 6 in the schematic (under the INITIAL NUMBERS OF OBJECTS). That means there is an open circuit somewhere. You can double click on the Nets in the RVE and they should be highlighted in the layout. Make sure that everything is connected as you think it should be.

You can also purposefully make a short circuit, change the "l" and "w" dimensions in schematic so that they dont match the "l" and "w" in the layout, change the device type and see the sort of errors you get in LVS, so that you get fimilar with them. (IMP - whenever you make any change to schematic, make sure you "Check and

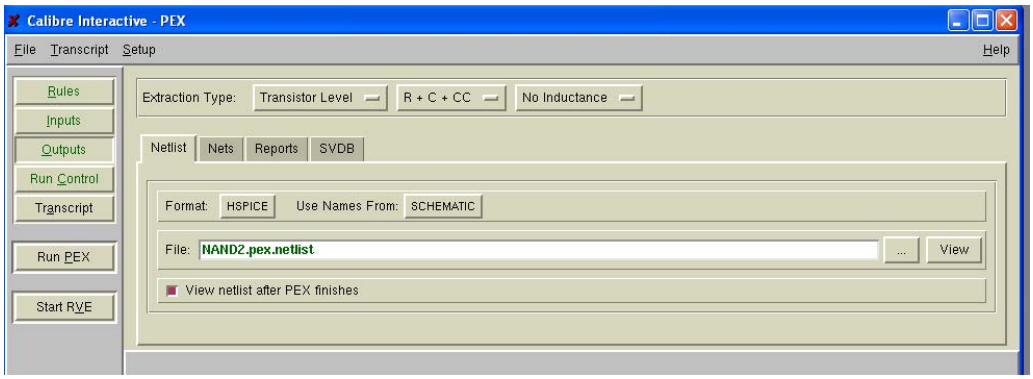
Save" and not just "Save", before you run LVS once again. If you dont do so, the LVS will through an error saying it cannot export the schematic)

In general, fixing LVS errors can be hard, so it's a good idea to add hierarchy to your layout (explained in the next tutorial) and keep things from getting too complex.

Extract Parasitics

Next, fix the layout of the nand2 gate and save the design. Now we're going to extract the parasitic wire capacitances and resistances from the layout.

To perform a Parasitic Extraction(PEX), choose **Calibre->Run PEX...**. The PEX form appears, as shown below.



Extract with Parasitic Capacitances and Resistances

There are a number of options you need to set and know what they are. For options not mentioned below, leave them as it is. We will use them in future if required.

Rules

Calibre-LVS Rules File and Calibre-LVS Run Directory are already filled in by the tool,leave them as it is.

Inputs

Under the **Layout** tab

Files : NAND2.calibre.gds

Top Cell: NAND2

These options are already be filled in by the tool, leave them as is

Format: select "GDSII" and select the option "Export from layout viewer" (This is very important)

Under the **Netlist** tab

Files: NAND2.src.net

Top Cell:NAND2

These options are already be filled in by the tool, leave them as is

Format: select "SPICE" and select the option "Export from schematic viewer" (This is very important)

Outputs

Extraction Type: Select Transistor Level, R + C + CC, No Inductance

Under the **Netlist** tab

File: NAND2.pex.netlist

This option is already be filled in by the tool, leave it as is

Format: HSPICE

Use Names From: SCHEMATIC

select "View netlist after PEX finishes"

Under the **Nets** tab

Extract parasitics for: Select "All Nets"

Under the **Reports** tab

PEX Report File: NAND2.pex.report

This option is already be filled in by the tool, leave it as is

Select "Generate PEX Report" and "View Report after PEX finishes"

Under the **SVDB** tab

SVDB Directory: svdb_nand2

Select "Generate cross-reference data for RVE" and "Start RVE after PEX"

Set the PEX form with the options shown above. Then click the “Run PEX” button. If PEX runs sucessfully, with out any error, then you will be able to view PEX Report File - NAND2.pex.report and also PEX Netlist File - NAND2.pex.netlist, which is the extracted netlist from the layout along with parasitic capacitances and resistances

.

Click on the "Transcript" tab in Calibre Interactive - PEX to see the log file.

Now let us understand the NAND2.pex.netlist File

The main hspice netlist "NAND2.pex.netlist" contains only the intentionally designed devices.
Filenames with the extensions ".pex" and ".pxi" files are included in the "NAND2.pex.netlist"

The **.pex** file (actually called "NAND2.pex.netlist.pex") contains one subckt per net: each subckt containing the RC tree structure modeling the net.

The **.pxi** file (actually called "NAND2.pex.netlist.NAND2.pxi") contains the connections between the parasitic networks i.e. containing the instance calls to the net model subckts along with the coupling capacitors connecting between these net model instances.

The NAND2.pex.netlist is as shown below

```
* File: NAND2.pex.netlist
```

```
* Program "Calibre xRC"
* Version "v2006.1_19.20"
*
.include "NAND2.pex.netlist.pex"
.subckt NAND2    GND! Z VDD! A B
*
* B          B
* A          A
* VDD!       VDD!
* Z          Z
* GND!       GND!
MM1 X N_A_MM1_g N_GND!_MM1_s N_GND!_MM1_b NMOS_VTL L=5e-08 W=9e-08
MM0 N_Z_MM0_d N_B_MM0_g X N_GND!_MM1_b NMOS_VTL L=5e-08 W=9e-08
MM2 N_Z_MM2_d N_A_MM2_g N_VDD!_MM2_s N_VDD!_MM2_b PMOS_VTL L=5e-08 W=9e-08
MM3 N_Z_MM2_d N_B_MM3_g N_VDD!_MM3_s N_VDD!_MM2_b PMOS_VTL L=5e-08 W=9e-08
*
.include "NAND2.pex.netlist.NAND2.pxi"
*
.ends
*
*
```

Extract with Parasitic Capacitances Only

Looking through the **.pex** and **.pxi** files, you should see that even this very simple circuit has resulted in hundreds of capacitors and resistors. This level of detail is necessary to achieve the highest degree of accuracy, but for larger circuits, it can slow an HSPICE simulation to a crawl. While the resistances are necessary to accurately understand the *delay* of a circuit, the *power* dissipation can be predicted very well using capacitances only. We often call this a "lumped-C model" for a net, as opposed to an "RC-tree model".

To extract a lumped-C model, make the following change in the *Calibre Interactive - PEX* window:

Outputs

Extraction Type: Change from "R + C + CC" to "C + CC"

Re-run PEX, and you should find that you have capacitances only. Not only that, but the number of capacitances has reduced by a factor of 10