SECTION OF ELECTRICAL ENGINEERING

EPFL STI – SEL | Téléphone : +4121 693 13 46
ELG | Fax :
Station nº 11 | E-mail : alexandre.levisse@epfl.ch
CH-1015 Lausanne | Site web : https://sti.epfl.ch/fr/sel/

Fundamentals of VLSI – project Full Custom, session 1

SEL October 2024

Full Custom Project
Session 4 : post layout simulation and optimization

1. PARASITIC EXTRACTION

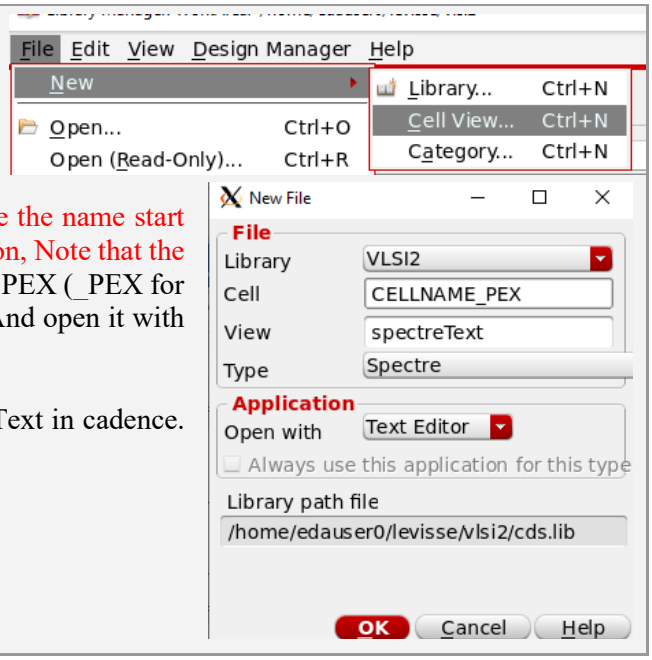
Work smart : Do not run post PEX simulations for each level, only run it on the top once you are done with the layout.

For the parasitic extraction, we noted that , while the approach proposed in LAB3 is correct and working, it can cause issues when repeated several times. We thereby propose here another approach, indeed less user-friendly but reliably PEX simulations.

The pex procedure is explained in the following. For the project, please follow these guidelines.

Once you have your finalized and verified layout (passing DRC and LVS checks), do the following :

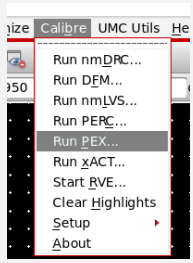
- ✓ First, create a cellview, in which you will host the extracted netlist. Select the cell that you want to extract and click New>Cell View…
- ✓ Give it a name that correspond to your current naming methodology (do NOT make the name start with a number, you may have troubles later on, Note that the names are cap-sensitive), with a reference to PEX (_PEX for e.g.). Make it a spectre type (spectreText). And open it with the text Editor.
- ✓ Click ok
- ✓ The new cellview now appears as a spectreText in cadence. Close it.



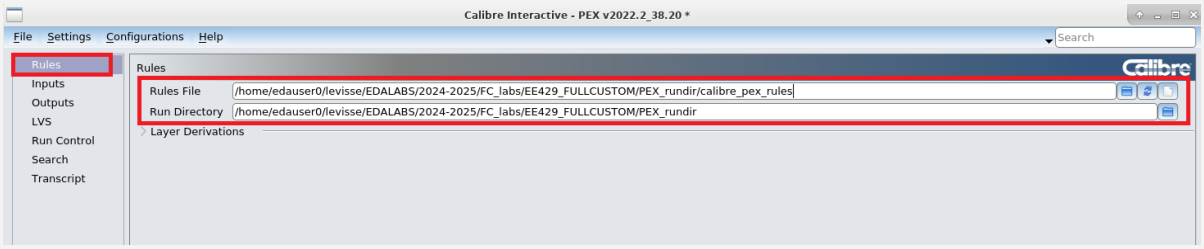
✓ Switch to the layout view of the cell you want to extract.

✓ Open Calibre>Run PEX

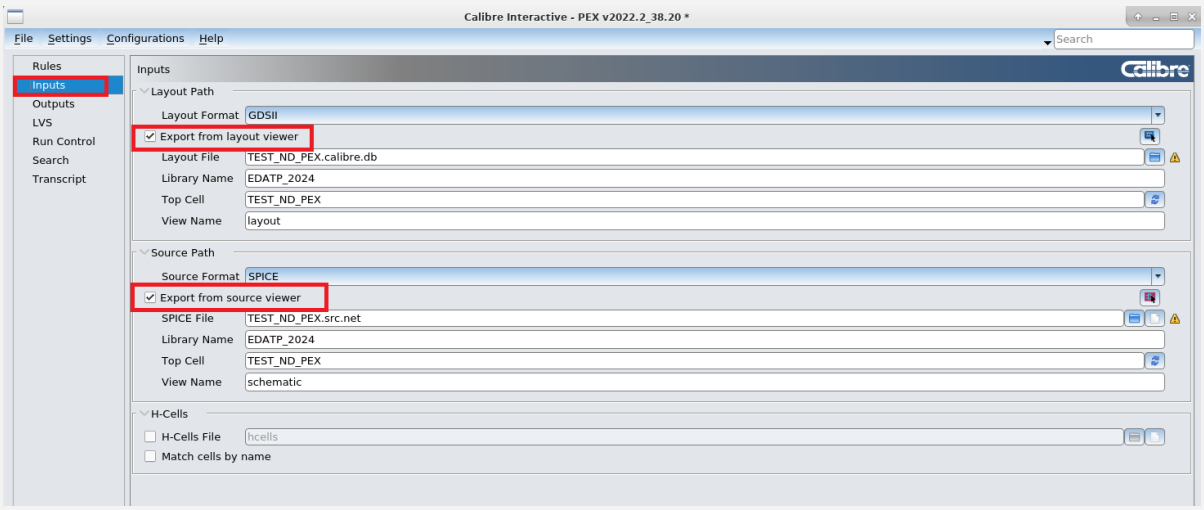
Parasitic extraction does something a lot alike the LVS, but then does generate a new netlist that can then be simulated and compared to the pre-layout netlist (schematic simulation).



✓ Select the **calibre_pex_rules** file inside the PEX_rundir folder. As for the LVS, select the Run directory as being the directory holding the rule file (in this example, PEX_rundir).

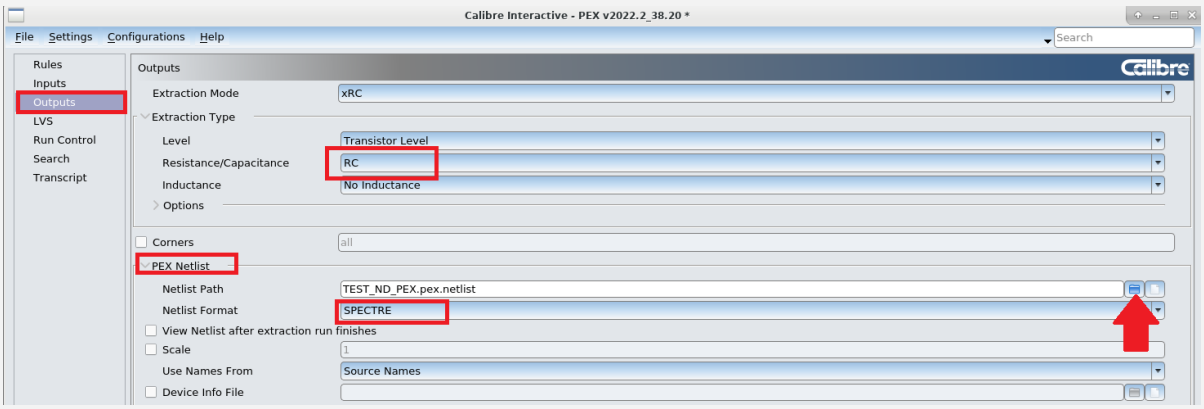


✓ In the Input page, under the layout tab, make sure that the box export from layout viewer is selected. Same for the netlist tab.

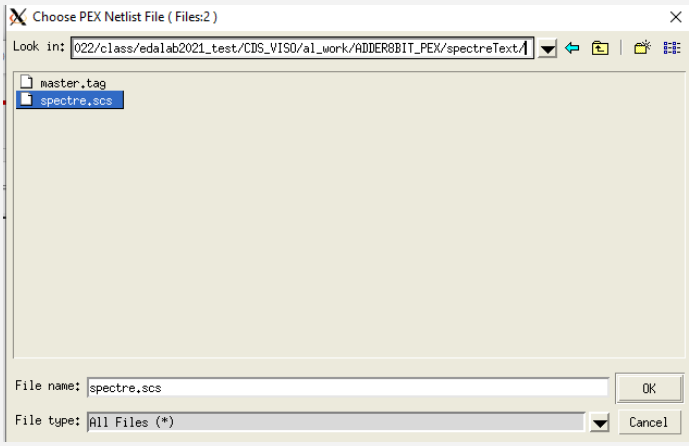


✓ In the Outputs page, in the netlist tab, make sure that format is set to SPECTRE and that you use name from SCHEMATIC.

✓ Set the extraction type to Transistor Level, and R+C, and No Inductance.



✓ In the File field, select the spectre.scs file that corresponds to the spectreText cell you just created inside virtuoso. In this example, the library is called "al_work" and the cell is "ADDER8BIT_PEX". Just chose the good path corresponding to your newly created cellview. Click OK. The tool will ask you if you are okay to overwrite the file. Click Yes.



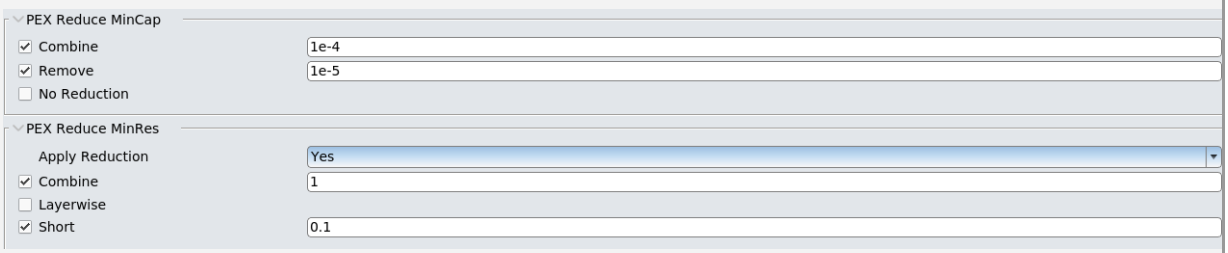
✓ To configure the resistance and capacitance filters:

Settings>Show Pages>Options

✓ Use the following parameters for the PEX runs in the Options panel:

In the **Reduction and CC** sub-tab, tick Enable **MinCap Reduction** and Enable **MinRes reduction**. Turn the MinRes to YES. Combine capacitances from 1e-4 pF and remove from 1e-5 pF. Combine resistances from 1ohm and short 0.1ohm.

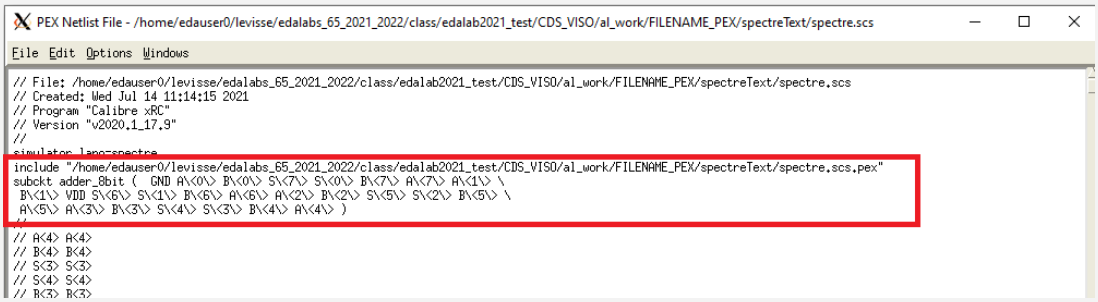
ℹ *This Option panel is really important. It allows you to filter the extracted capacitances from your design to limit the size of your netlist. It is up to the designer to identify good values to filter capacitances. Filtering to too high values will accelerate the simulation but degrade the precision.*



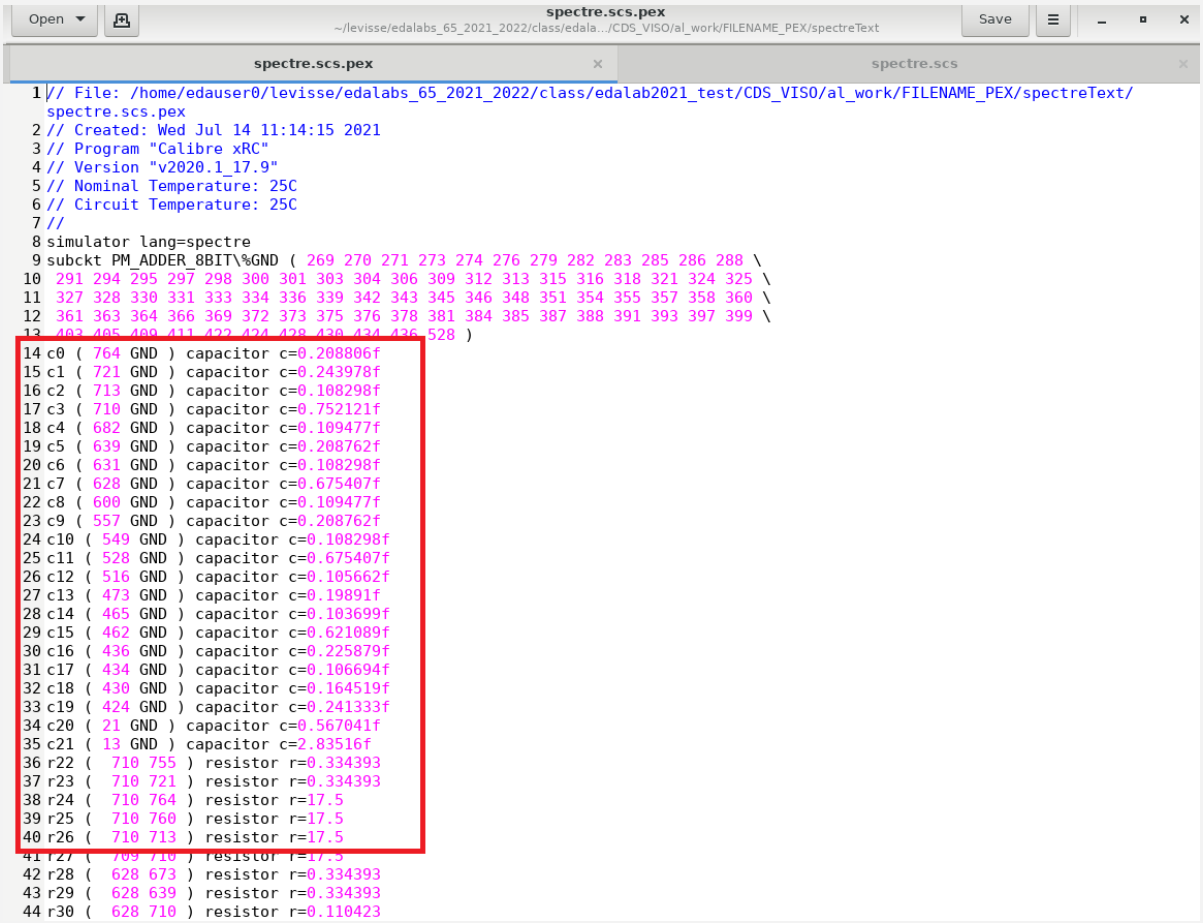
✓ Click on Run PEX

ℹ *If you ticked the "View netlist after PEX finishes"on the output panel, it will show you the extracted netlist at the end of the process. Spectre netlists are not generated in a single file. The main file defines the transistors and nets.*

ℹ The Spectre.scs.pex file contains all the parasitics elements.



Note that there are no resistance and capacitances values lower than the values you define as limits in the calibre PEX options.



✓ Close the window and open the netlists with a text editor from your terminal (you can also open the spectre.scs netlist from the library manager window, but the editor is not really convenient to use).

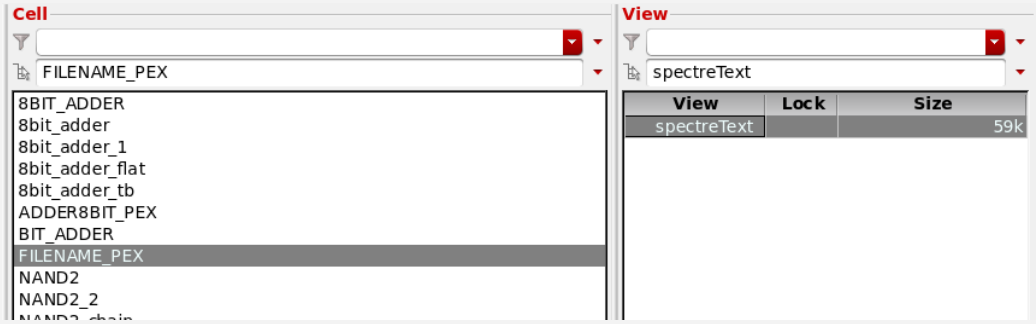
[edauser0@selsrv1 CDS_VISO]$ gedit EDATP/FILENAME_PEX/spectreText/spectre.scs &

✓ Correct the name of the subckt created by Calibre PEX with the name of the cell at the beginning (line 8) and at the end of the file. You could also do this step by opening the spectreText view in the library manager.

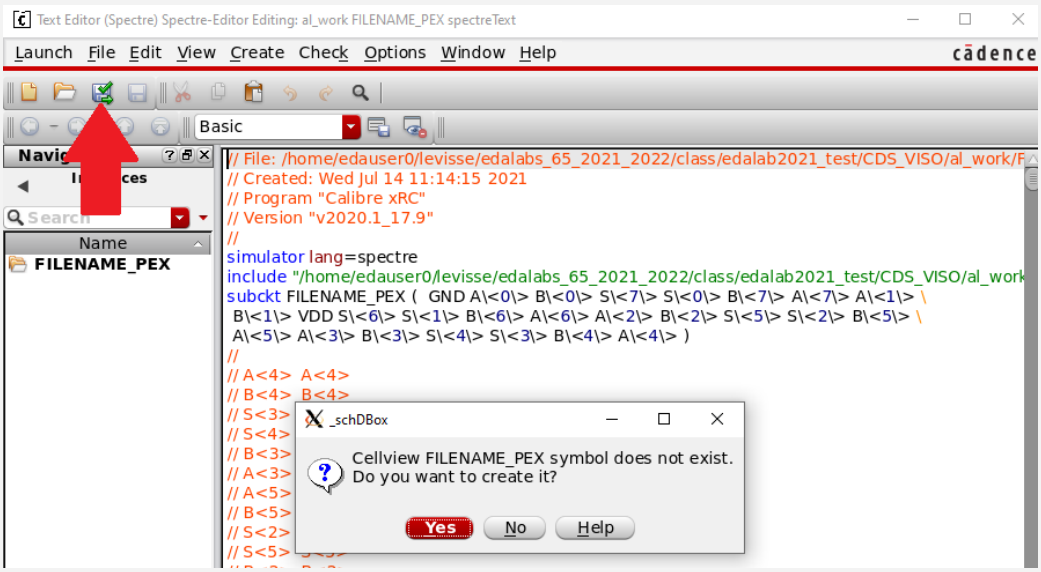
Using gedit, take a look at the other files that are being included inside the spectre.scs file. What do you see there ? ***do not modify them !***

```
 8 subckt FILENAME_PEX (  GND A\<0\> B\<0\> S\<7\> S\<0\> B\<7\> A\<7\> A\<1\> \
 9  B\<1\> VDD S\<6\> S\<1\> B\<6\> A\<6\> A\<2\> B\<2\> S\<5\> S\<2\> B\<5\> \
10  A\<5\> A\<3\> B\<3\> S\<4\> S\<3\> B\<4\> A\<4\> )
11 //
931  N_VDD_XI7\/MPM13_b ) P_12_LLRVT l=6e-08 w=2.4e-07 ad=3.84e-14 as=2.4e-14 \
932  pd=8e-07 ps=4.4e-07 nrd=0 nrs=0 sa=1.6e-07 sb=1.35e-06 sca=16.52 scb=0.019587 \
933  scc=0.000974964
934 //
935 include "/home/edauser0/levisse/edalabs_65_2021_2022/class/edalab2021_test/CDS_VISO/al_work/FILENAME_PEX/spectreText/
    spectre.scs.ADDER_8BIT.pxi"
936 //
937 ends FILENAME_PEX
938 //
```

✓ Save the file and close. If you used gedit to edit the netlist, go back to your library manager. Select your cellview as shown in the next screenshot. Double click on the spectreText to open it.



✓ Click on the "**Build a database of instances, nets and pins found in file**" button. 

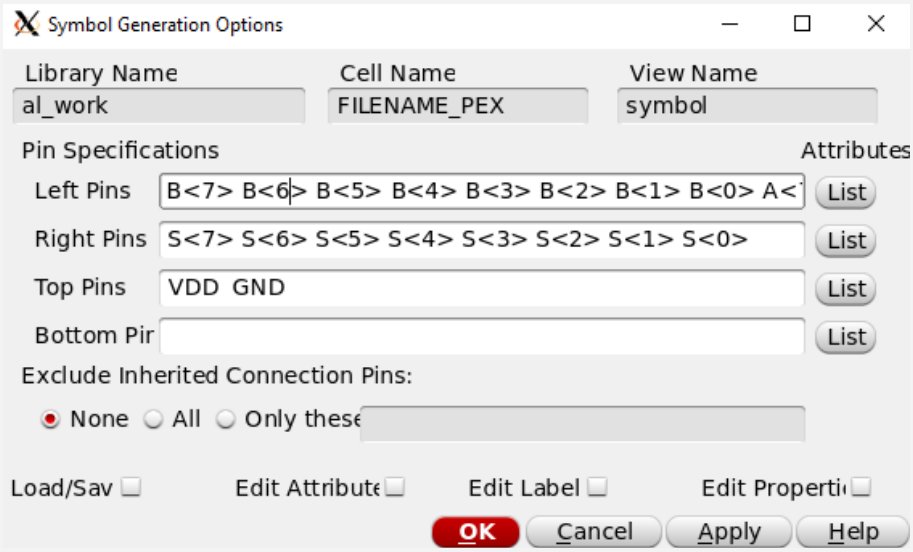


✓ At this point, the tool will ask you if you want to create a symbol for your cellview. Click yes.

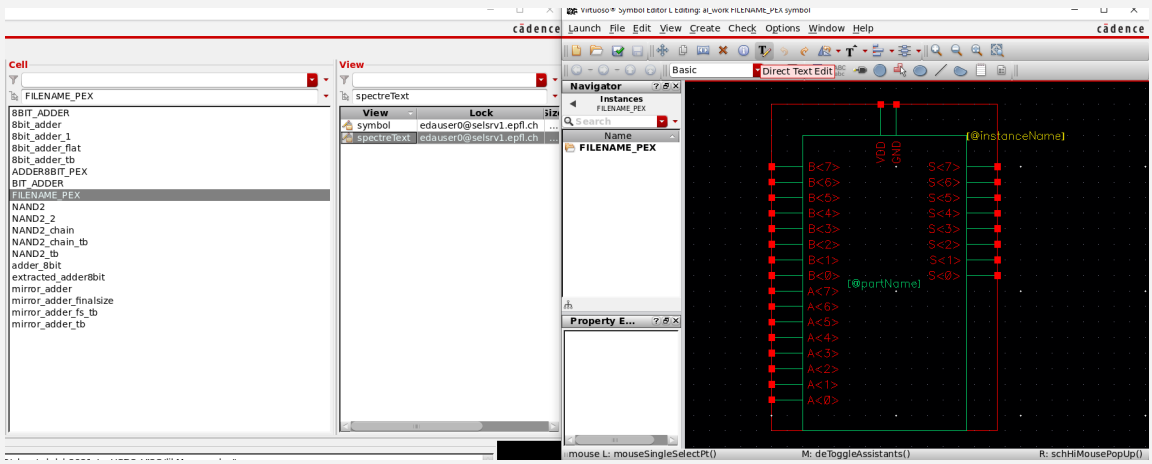
If the tool tells you that you have a syntax error, check that the name you gave for the subckt at the beginning end at the end the file is the same than the cellview name. if you used gedit to modify the file, make sure that the file is not being opened/locked in virtuoso when saving it. Also, make sure that your subckt name does not start with a number and that you respect the capital letters in both the name and cellview name. This will make the simulator/parser crash. Finally, if you renamed the cellview, make sure that you updated the paths associated with the

"include" statements to account for the new name. If none of these solutions work, try to follow the flow again with a new cellview. ***Then, if none of this works,*** *you can call a TA.*

✓ Create the symbol as shown in the screenshot. A good practice is to keep input on the left, outputs on the right. For the postPEX symbol creation, do **not** merge the pins (e.g. A<7:0>). Click OK.



✓ A new symbol is not being created. Check in the library manager that you now have a "symbol" view. Save and close it. The example in the screenshot is for a 8bit adder.



✓ Call a TA to check your circuit with you.

ℹ Here, you will note that the calibre PEX tool does create as many pins as there are wires. You can select the instance and press "space" to avoid naming them by hand. And then, with the "l" key, create a bus label (e.g., A<0:7>) that you attach to a single wire, to make all these wires a bus again. Also, note that you could create a wrapper cellview in which the wires are connected to a bus.

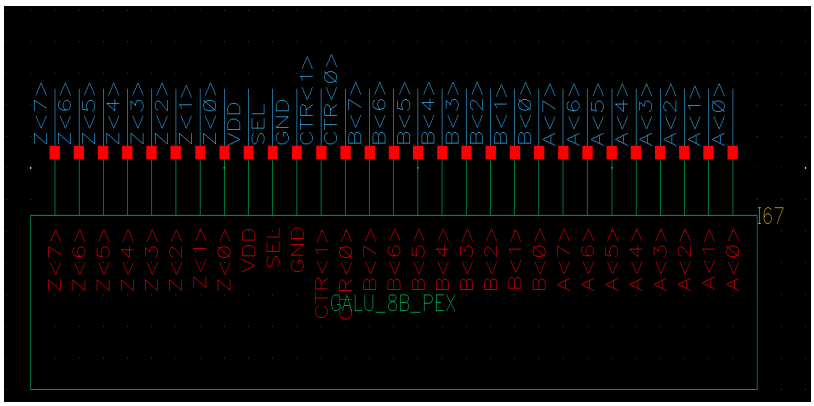
The post PEX simulation is then the same as what you did in LAB3 section 3.6.

- Import both the pre and post PEX symbols
- Adapt the names and capacitances
- Adapt the environment list
- Run the simulation

2. RE-OPTIMIZATION AND FINAL VERIFICATION

Run a simulation with the post PEX netlist and characterize the critical paths as you did in lab 3. Do not forget to add “spectreText” in the first position of the Switch View List in the environment options.

e.g. of a final ALU symbol after the PEX process. There is no constraint on the shape of the symbol. The names MUST correspond.



Hint : to create all the labels with the correct names, select the cell, and press the spacebar.

You should expect a post PEX degradation ranging from 30% to 50% compared to your schematic simulation. If it is worse than 60%, you may want to check the following parameters in your layout :

- length of the polysilicon lines. Try not having them too long. Longer than 700nm-1um may be an issue.
- Length of ME1 lines.
- Vias are resistive. And may add a lot of parasitic resistance to your signals. Doubling them can sometimes help.
- Long wires driven by small drivers

3. DELIVERABLE PREPARATION

Fail/Pass criterion :

- The design must work below 1ns of period post PEX
- The design must be smaller than 500um2
- The design must have a correct functionality
- No DRC/LVS errors

Additional metrics:

- Faster designs get a better grade
    - We apply a delay*area product to evaluate them
- Floorplan quality/justification
- Good practice in terms of routing, power management, substrate biasing

3.1. REPORT DELIVERY

Fill the powerpoint with your timing, area metrics as well as your screenshots as requested. You can submit it either as a PDF or PPTX format. **Do not use any other format**.

➔ The reference PPTX can be found on moodle.

3.2. ARCHIVE OF THE DESIGN LIBRARY

The file EDAUSER_X.tar.gz shall include the full EDATP library (X = your EDAUSER number). To create this file, follow these instructions :

1. close virtuoso
2. from your terminal, go in your EE429_FULLCUSTOM folder
3. Run the following command : **tar -zcvf EDAUSER_X.tar.gz EDATP** where X is your edauser number (this command will create an archive in your EE429_FULLCUSTOM folder that you can then upload on moodle).
4. submit the EDAUSER_X.tar.gz archive on moodle

Please note that in order to upload the file from the servers, you have to launch Firefox from the Linux terminal (simply type: firefox). Then you will be able to login on the moodle and complete the submission.