SECTION OF ELECTRICAL ENGINEERING

EPFL STI – SEL
ELG
Station nº 11
CH-1015 Lausanne

Téléphone : +4121 693 13 46
Fax :
E-mail : alexandre.levisse@epfl.ch
Site web : https://sti.epfl.ch/fr/sel/

Fundamentals of VLSI – Full Custom project
SEL October 2025

Full Custom Project
Use the automated test to verify your circuit

only use this flow once you have designed your circuit. This simulation flow will not give you the critical paths but help you identify some that you may have missed.

Your grading will be based off the results given by this testbench.

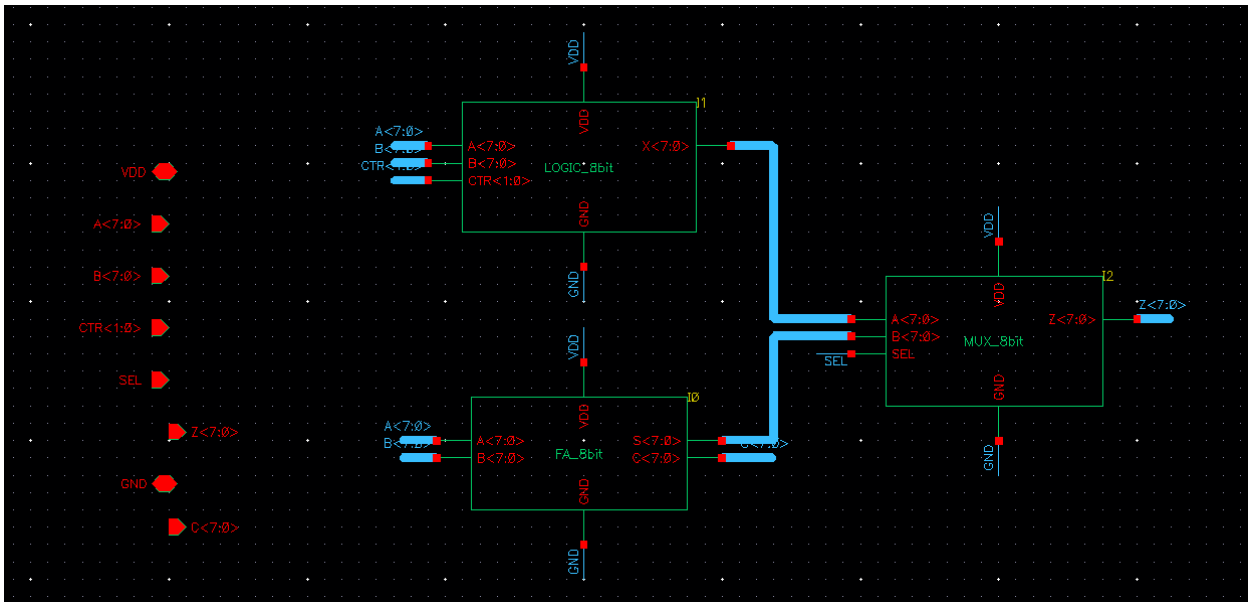
1. SCHEMATIC SIMULATION

1.1. CREATE THE NETLIST

open your top schematic.

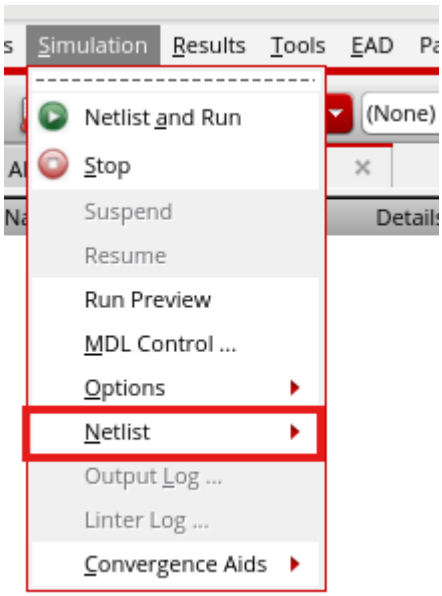
It is of primary importance that the following conditions are met :

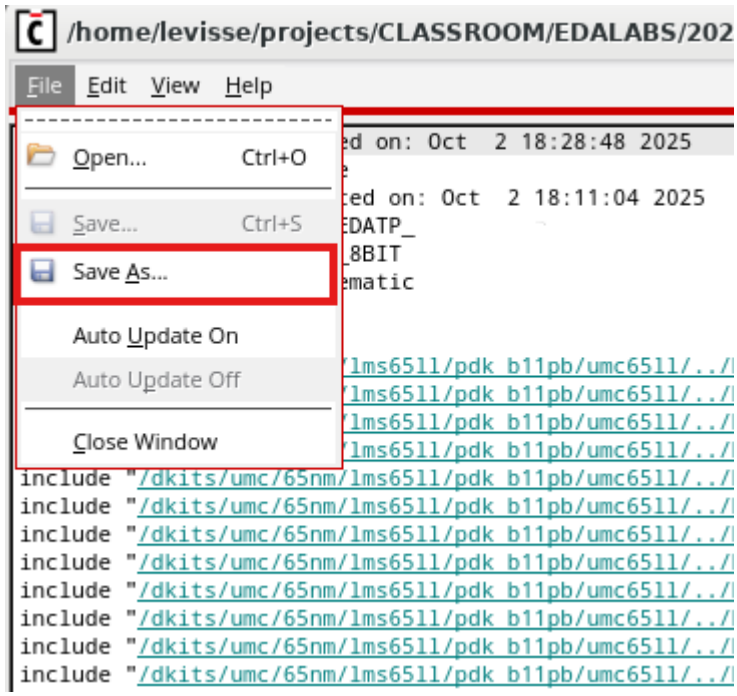
- all the signals are called EXACTLY A<7:0>, B<7:0>, Z<7:0>, SEL, CTR<1:0>
- you have pins with the correct names
- your circuit is checked and saved



Click Launch>ADE Explorer , and select create a new view

This will open an ADE explorer view, from which on the top panel, you click on Simulation>Netlist>Create





Then click on File>Save As… create, in your working directory, a folder called NETLISTS.

And in this folder save the netlist with the following naming convention: alu8bit_schematic_date_time.scs

Where date is today's date, and time the current time. For e.g. if it is the 12th of October at 15h35, your netlist will be called alu8bit_schematic_12oct25_1535.scs.

This is a good way to avoid overwriting unintentionally files or forget what was done when. Many methods exist for this purpose.

1.2. RUN THE SIMULATION

go back to your terminal. From your working directory, run the following command:

>/education/classes/2025-2026/EE429/PROJECT/verify_my_circuit/run_simulation.csh SCHEMATIC ./NETLISTS/alu_8bit_schematic_date_time.scs

This will run a simulation that tests

1- the functionality of your circuit
2- the timing of your circuit

scroll up in the logs and read them.

Here is an example of a correct simulation for two different patterns.

You can use this information to identify which paths of your circuits are too slow. And adapt accordingly.

A[7:0] = 0 0 0 0 0 0 0 0 , B[7:0] = 1 1 1 1 1 1 1 1 , CTR[1:0] = 0 0 , SEL = 1 , at 4.4 ns
Z[0] rises or falls after 107.772ps
Z[5] rises or falls after 108.124ps
Z[2] rises or falls after 108.469ps
Z[1] rises or falls after 108.626ps
Z[3] rises or falls after 120.985ps
Z[4] rises or falls after 123.773ps
Z[6] rises or falls after 564.493ps
Z[7] rises or falls after 603.294ps
A[7:0] = 0 0 0 0 0 0 0 0 , B[7:0] = 0 0 0 0 0 0 0 0 , CTR[1:0] = 0 0 , SEL = 1 , at 5.9 ns
Z[6] rises or falls after 57.0385ps
Z[7] rises or falls after 61.221ps
Z[3] rises or falls after 79.0374ps
Z[4] rises or falls after 79.1825ps
Z[1] rises or falls after 87.4469ps
Z[2] rises or falls after 92.3816ps
Z[0] rises or falls after 104.48ps

```
Z[5] rises or falls after 105.621ps
```

Here is an example of an incorrect simulation:

```
A[7:0] = 0 0 0 0 0 0 0 0 , B[7:0] = 0 0 0 0 0 0 0 0 , CTR[1:0] = 1 0 , SEL = 0      , at 80.9 ns
Z[6] rises or falls after 63.8164ps
Z[4] rises or falls after 63.8746ps
Z[5] rises or falls after 63.9011ps
Z[0] rises or falls after 63.9861ps
bit 0 -----------------ERROR CODE---------------------- at 82.4 ns
bit 7 -----------------ERROR CODE---------------------- at 82.4 ns
```

At the end of the simulation, it tells you how many functionality errors exist, and the slowest path in the design :

```
8 errors in the design

slowest switch is 608.014ps at 21.6105ns
```

2. POST PEX SIMULATION

The method is highly similar to schematic simulation.

For this flow, you do not need to rename the top cell view, as this is not done through virtuoso.

2.3. RUN THE PEX

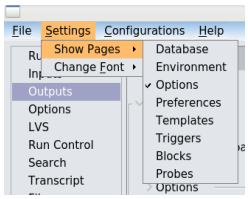
Make sure you run the PEX from a cellview called extractly “ALU_8BIT” otherwise this will not work

- Remember that your topview shall be called ALU_8BIT, note that cadence tools and simulators are case sensitive.
- Your design shall have exactly the correct number of pins as expected by the testbench A<7:0>, B<7:0>, Z<7:0>, SEL, CTR<1:0>

2.3.1. TO ENSURE THAT THE ORDER OF THE PINS GENERATED IS CORRECT :

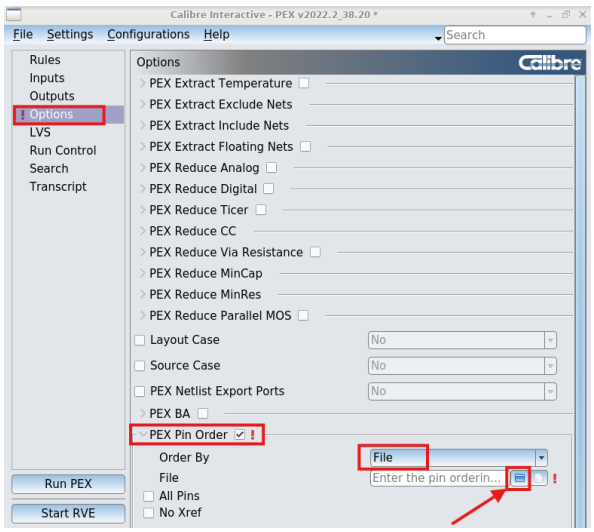
when running PEX, calibre is generating the pin order based on the extraction from schematic or layout. The grading testbench expects a certain pin order. We use a template to force it to a format that the automated testbench can understand.

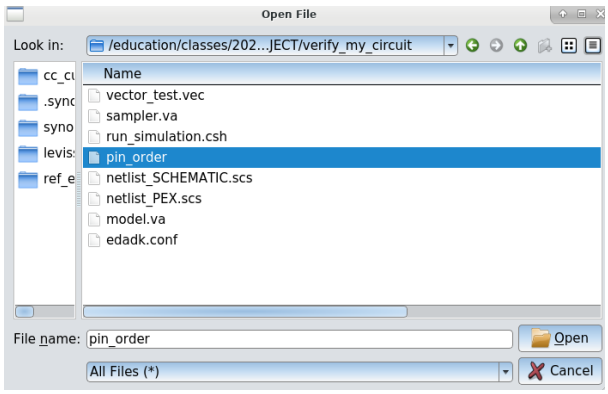
In calibre when running PEX, in the PEX options



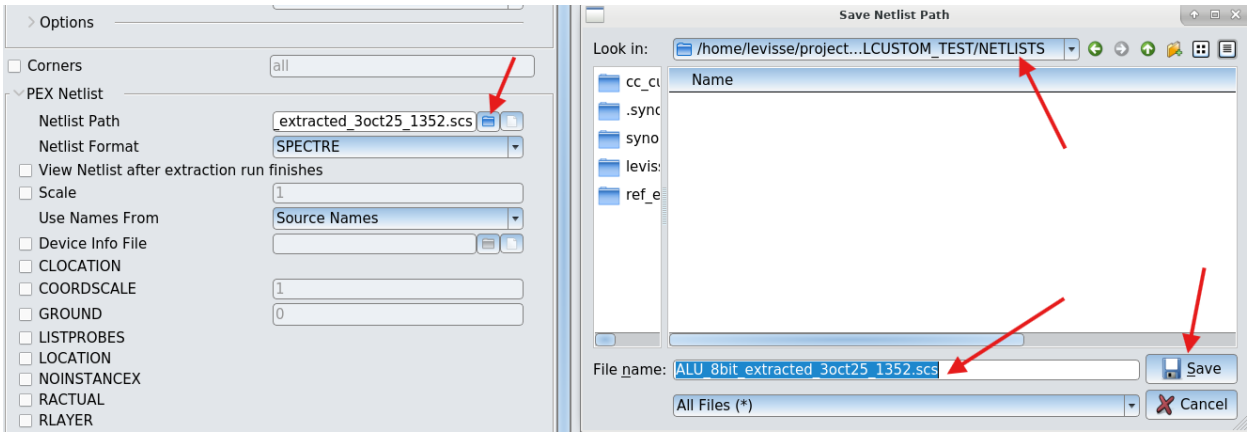
In the “PEX Pin Order section”, select File and point to the “pin_order” file in

/education/classes/2025-2026/EE429/PROJECT/verify_my_circuit/





2.3.2. SAVE THE NETLIST IN YOUR NETLISTS FOLDER



call it ALU_8BIT_extracted_date_time.scs

2.4. RUN THE SIMULATION

Here simply provide your post PEX spectre netlist generated with calibre to the script.

Example :

>/education/classes/2025-2026/EE429/PROJECT/verify_my_circuit/run_simulation.csh PEX ./NETLISTS/ALU_8BIT_extracted_date_time.scs

Note that here the second argument in the command is “PEX” and not “SCHEMATIC”

3. ANNEX FOR THE MOST COMFORTABLE STUDENTS

during PEX simulation, the automated test instantiates the following cellview :

```
I9 (CTR\<0\> CTR\<1\> VDD GND SEL B\<7\> A\<7\> B\<6\> A\<6\> B\<5\> A\<5\> B\<4\> A\<4\>
B\<3\> A\<3\> B\<2\> A\<2\> B\<1\> A\<1\> A\<0\> \
 B\<0\> Z\<7\> Z\<6\> Z\<5\> Z\<4\> Z\<3\> Z\<2\> Z\<1\> Z\<0\>) \
      ALU_8BIT
```

Your spectre top cell must match this structure with its name and pin order.

If you want to run your PEX from a cellview that has a different name than “ALU_8BIT”:

- make a copy of the pin_order file, duplicate the ALU_8BIT subckt and rename it (there can be as many references as you need in that file). This will properly order the pins for your cellview.
- In the extracted netlist, rename the top view to call it ALU_8BIT on the first, and last line.