



Exercise 2: Designing with (Schematic) Hierarchy

Prof. Dr. P. Fischer

Lehrstuhl für Schaltungstechnik und Simulation
Uni Heidelberg



Concept

- Schematics can contain
 - Library elements
 - Wires, pins, ...
 - **Symbols** representing other cells

- Therefore a schematic view should also have a *matching symbol view*
 - It must have the *same pins*
 - *Directions of all pins* must be the same

- The ‘graphics’ of the symbol should be clear.
 - inputs left
 - outputs right
 - ...

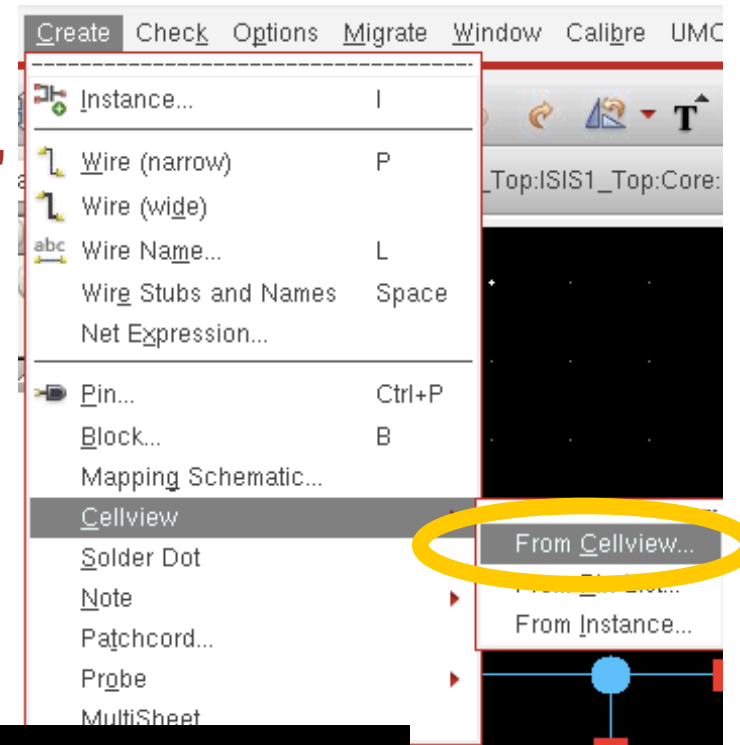


Implementation

1. You can create a *schematic* and a *symbol* view in the same *cell* independently.
 - When you save either cell, the number/names/types of the pins are checked against the other cell.

2. You can also create the 'second' view *automatically*: Create → Cellview → From CellView

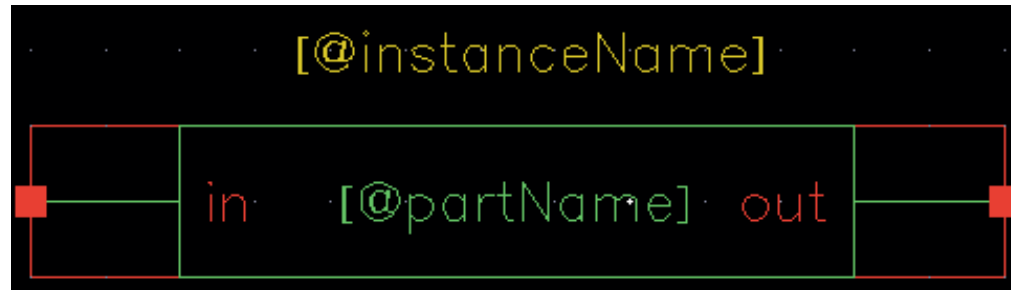
- Two further forms are displayed..
- This works in both directions
 - Symbol → Schematic
 - Schematic → Symbol



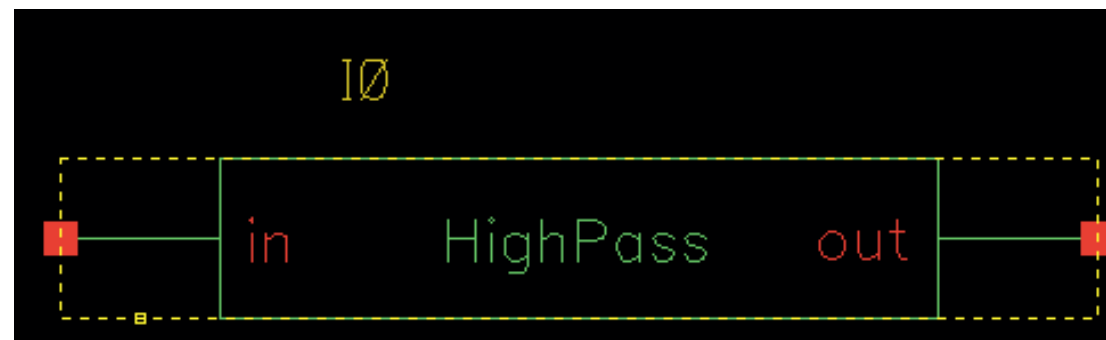


partName and instanceName

- When you automatically create a symbol, you see two text objects:



- When you instantiate the symbol, these objects are replaced by meaningful content:
 - `[@partName]` is replaced by the Name of the cell (in the library)
 - `[@instanceName]` is replaced by the name of the instance

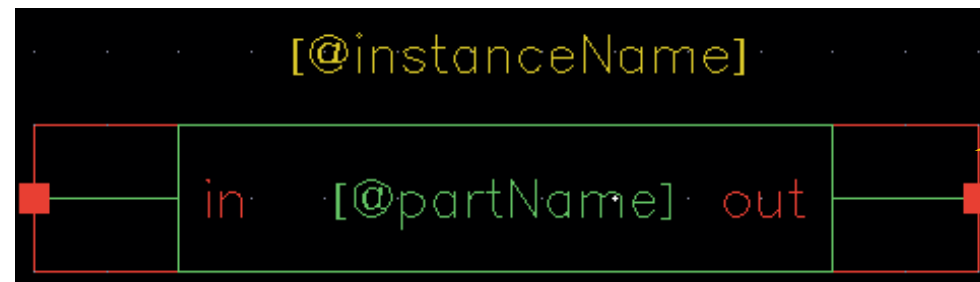


- Position, size, alignment can be changed (do NOT change text!)



The Selection Box

- When created automatically, a (red) *Selection Box* appears
- It marks the area which will be used to 'highlight' / 'select' the instance (in the next hierarchy level):



Selection Box
in **symbol**



Highlighting in a
schematic which uses
an instance of the cell

- The *Selection Box* can be moved / resized
- If lost (or in manually created cells), it can be created by **Create** → **Selection Box**
- You cannot route over the Selection Box → keep it small
- If no Selection Box is defined, the maximal symbol size is used.

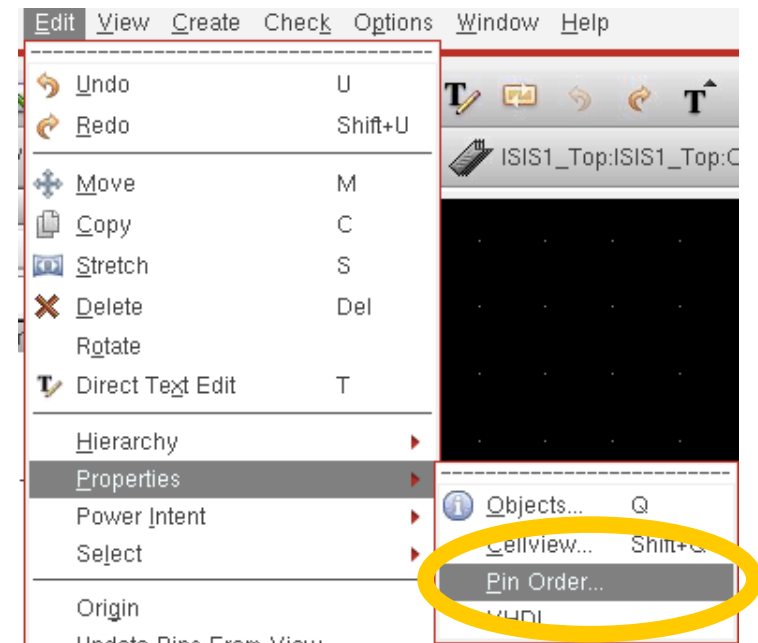
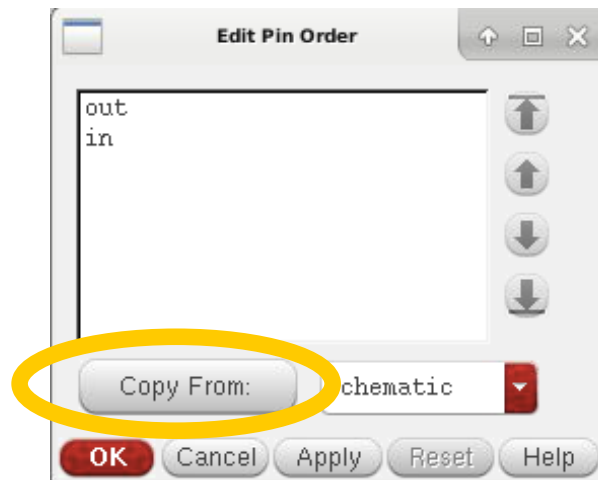


(Pin Order)

- If can happen that the internal order of pins gets messed up
 - You get a warning at Check & Save
 - This happens if you copy pins from other cells, delete of pins,..

- To restore correct order, use **Edit → Properties → Pin Order**

- Best copy the Pin Order from another view:



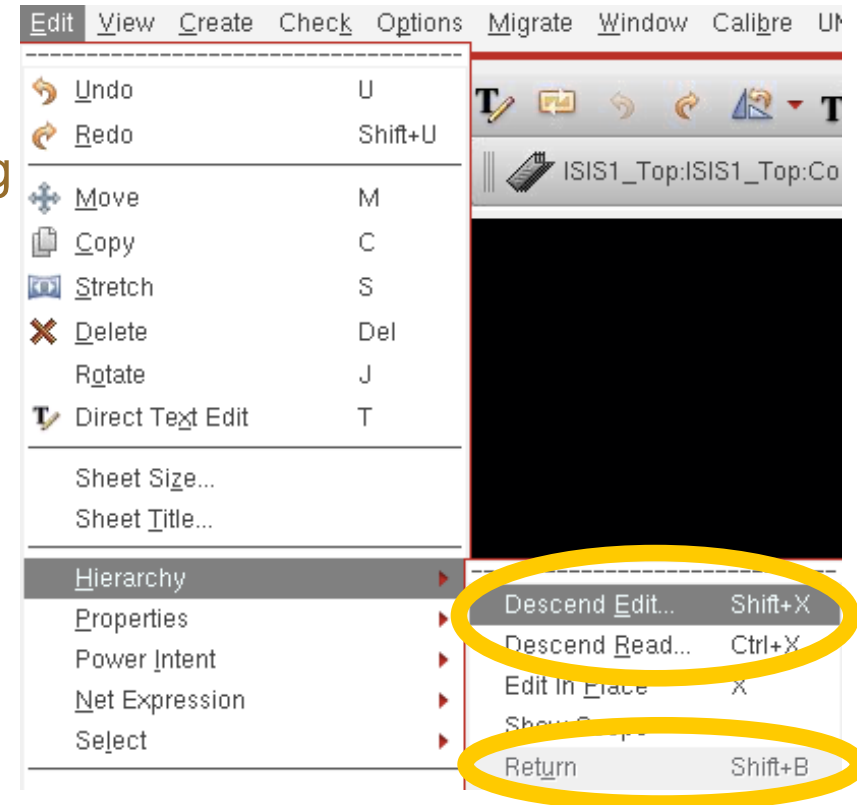


Traveling down/up in the Hierarchy

- If you want to inspect the schematic which belongs to (the symbol of) an instance:

- Select the instance
- 'Dive' 'down' into the cell using **Edit → Hierarchy → Descend** or with key **Shift-X**

- Return back 'up' with **Shift-B**



- **Powerful & Risky:**

- You can 'dive' into the cell (go down the hierarchy) but still display the *schematic*. This allows you to immediately see your changes (in all instances of the cell type)
- Do NOT forget to go back up!

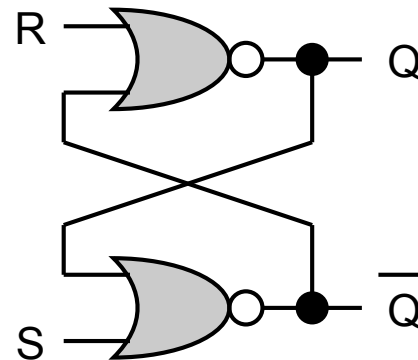


EXERCISE 4A: A SET-RESET FLIPFLOP



Creating the Cells

- A Set-Reset Flipflop (SF-FF) can be implemented by the following schematic



1. Create Schematic and Symbol of a NOR2

- Use global nets vddd! and gnd! for supply / ground
- Use two input pins I0, I1 and one output pin Q
- Use minimum size NMOS (N_18_MM) and PMOS (P_18_MM) from the UMC library UMC_18_CMOS

2. Create Schematic and Symbol of the SF-FF

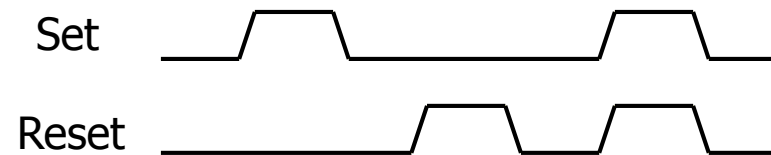
- Use pin names as shown above
- Instantiate two of your NOR2 gates



Simulation

3. Create a simulation schematic and simulate the SR-FF

- Do not forget the voltage supply source (use 1.8V)!
- Add *vpulse* sources from *analogLib* to generate stimuli, for instance:



(the pulses should be a few Nanoseconds long)

- Is everything as expected?
- What is the delay between inputs and Q and !Q?
- What happens if Set=Reset=1?

- Add capacitive loads (1pF) and see what happens!