RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

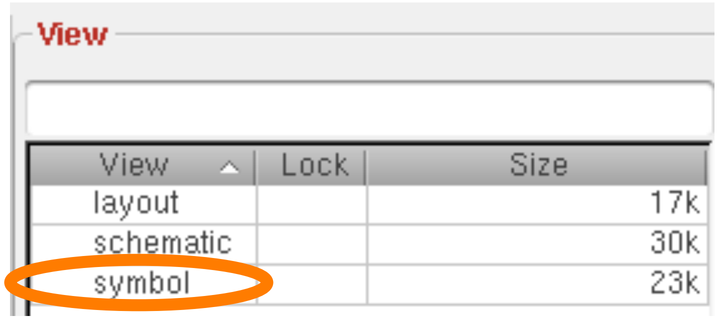
Schematics

Prof. Dr. P. Fischer

Lehrstuhl für Schaltungstechnik und Simulation
Uni Heidelberg

What are Symbols ?

- Very often, a circuit (schematic) can be re-used.
- Instead of copying everything, we can ‘include’ the schematic into another schematic
- In order to identify the nets, we need a ***symbol***
  - This is a new ***view*** type

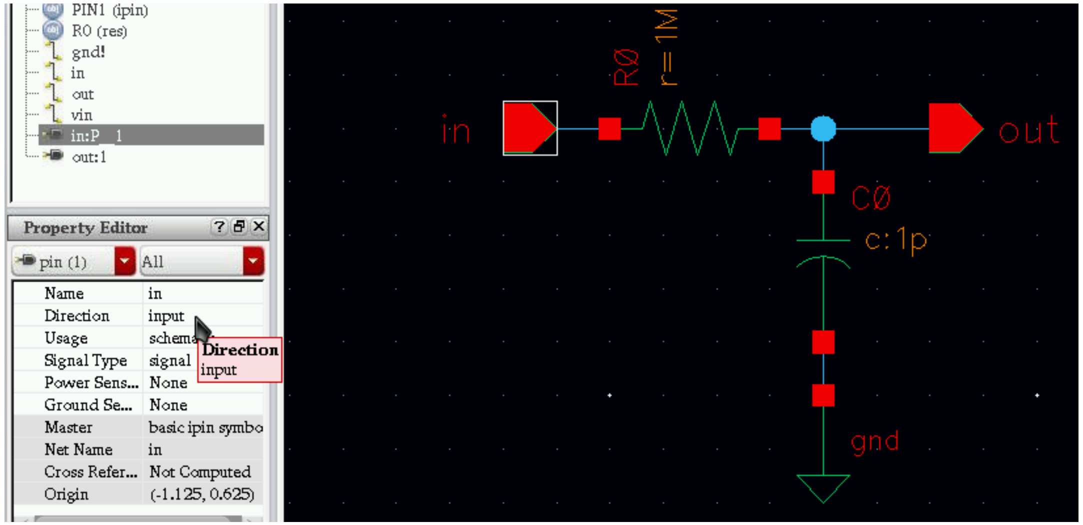


- The nets which should be passed to the outside world must be connected to ***pins*** in the schematic.
- For ***each pin in the schematic*** we also need *a* ***pin in the symbo****l*.

- Pins have the same name as the connected net
- They can be ***input / output / inputOutput*** (see later)

Preparing the Schematic

- The easiest way to create a symbol starts from a schematic
- Using Create → Pin (Ctrl-P or button ), create pins for all signals that should be visible 'outside'
  - **outputs** are signals that will drive to other cells
  - **inputs** only receive signals. They **must** be connected
  - InputOutput are most general

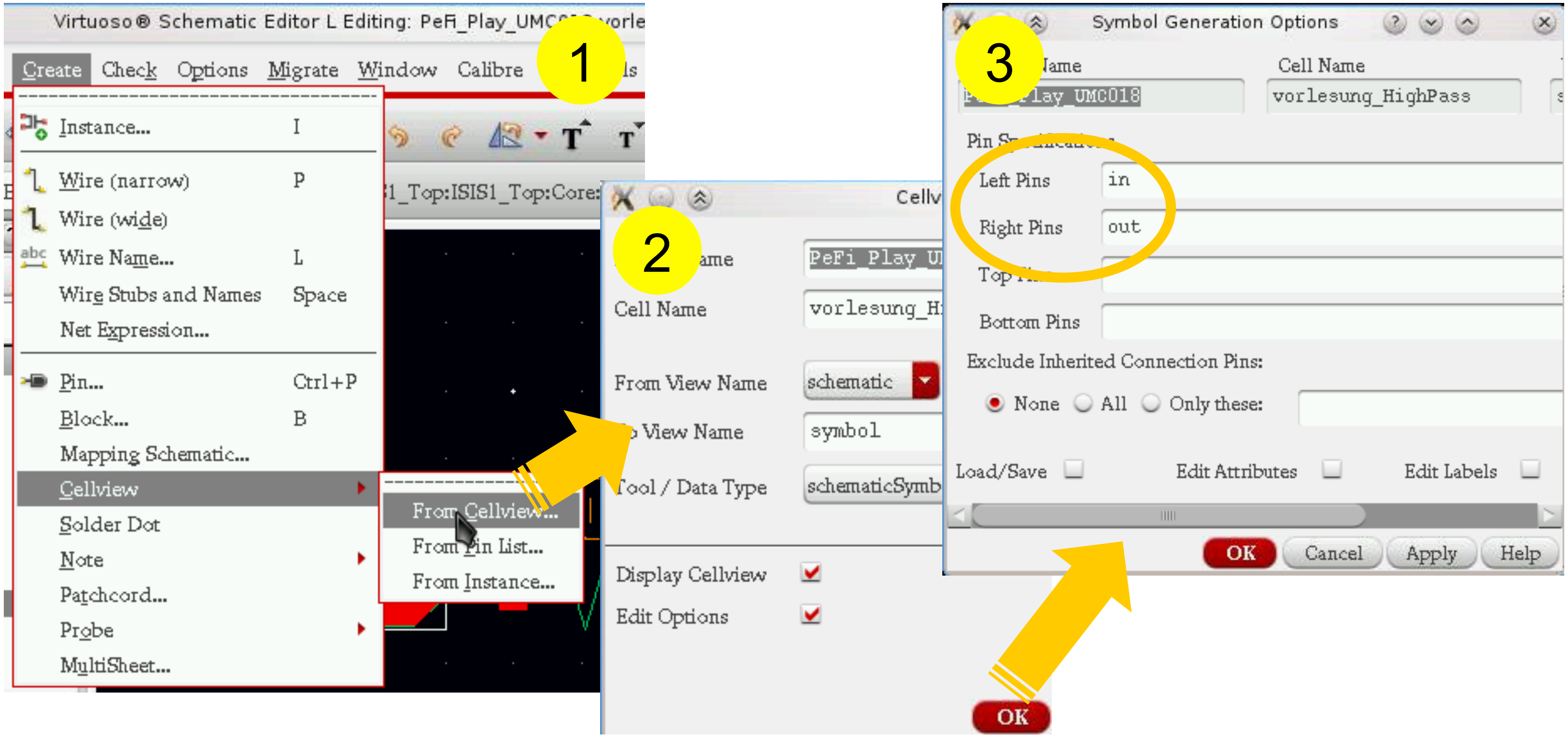


- Better remove all symbols used for simulation (sources..)

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Creating a Symbol from the Schematic

1. Select Create → Cellview → From Cellview
2. Check that ‘From View’ is *schematic* and ‘To view’ is *symbol*
3. Press ok. In the next window, select pins are locations



Editing the Symbol

- A symbol template is created:

A pin. Here: Input

Name	in
Direction	input

This special label will later be replaced by the name of the master cell

This special label will later be replaced by the name of the INSTANCE

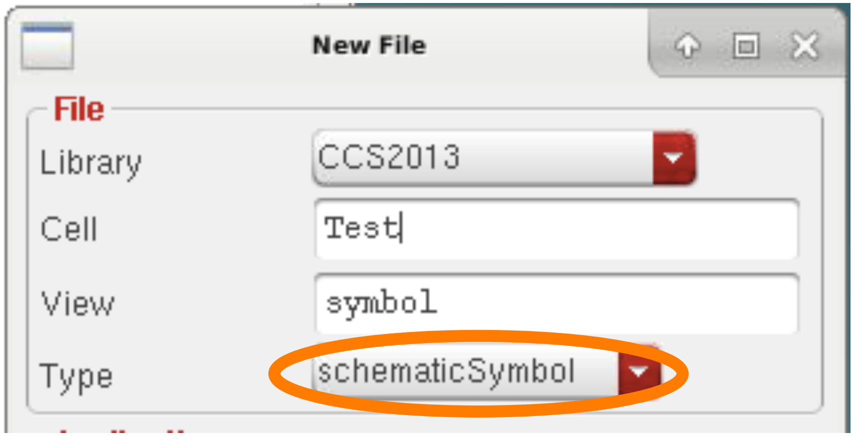
Green lines are only for optics, they have **no function**. You can add more such shapes in the Create → Shapes menu

The red BOX is the '**selection box**'. It is the region where you can later select the symbol. (I prefer to keep it small…)

- You can set the origin under Edit → Origin

Creating a Symbol from Scratch

- You can also create a new symbol directly from the library browser with File → New → Cell View… with view type ***schematicSymbol***

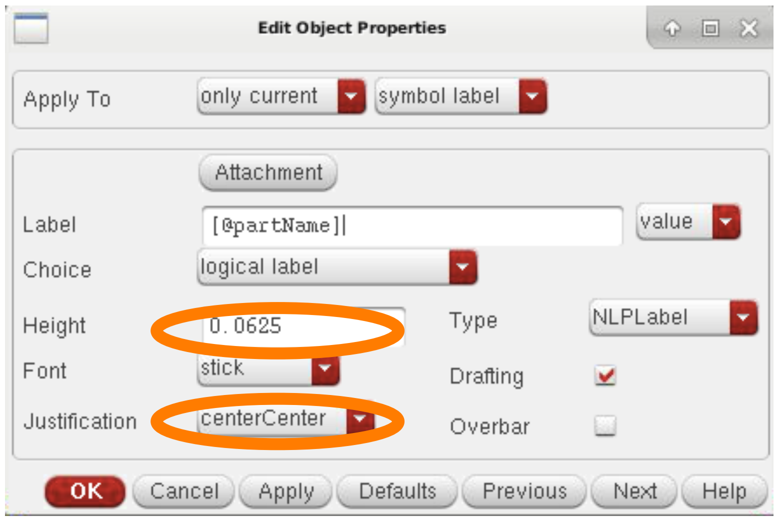


- You must then place all pins, boxes, labels, .. by hand.

Editing a Symbol

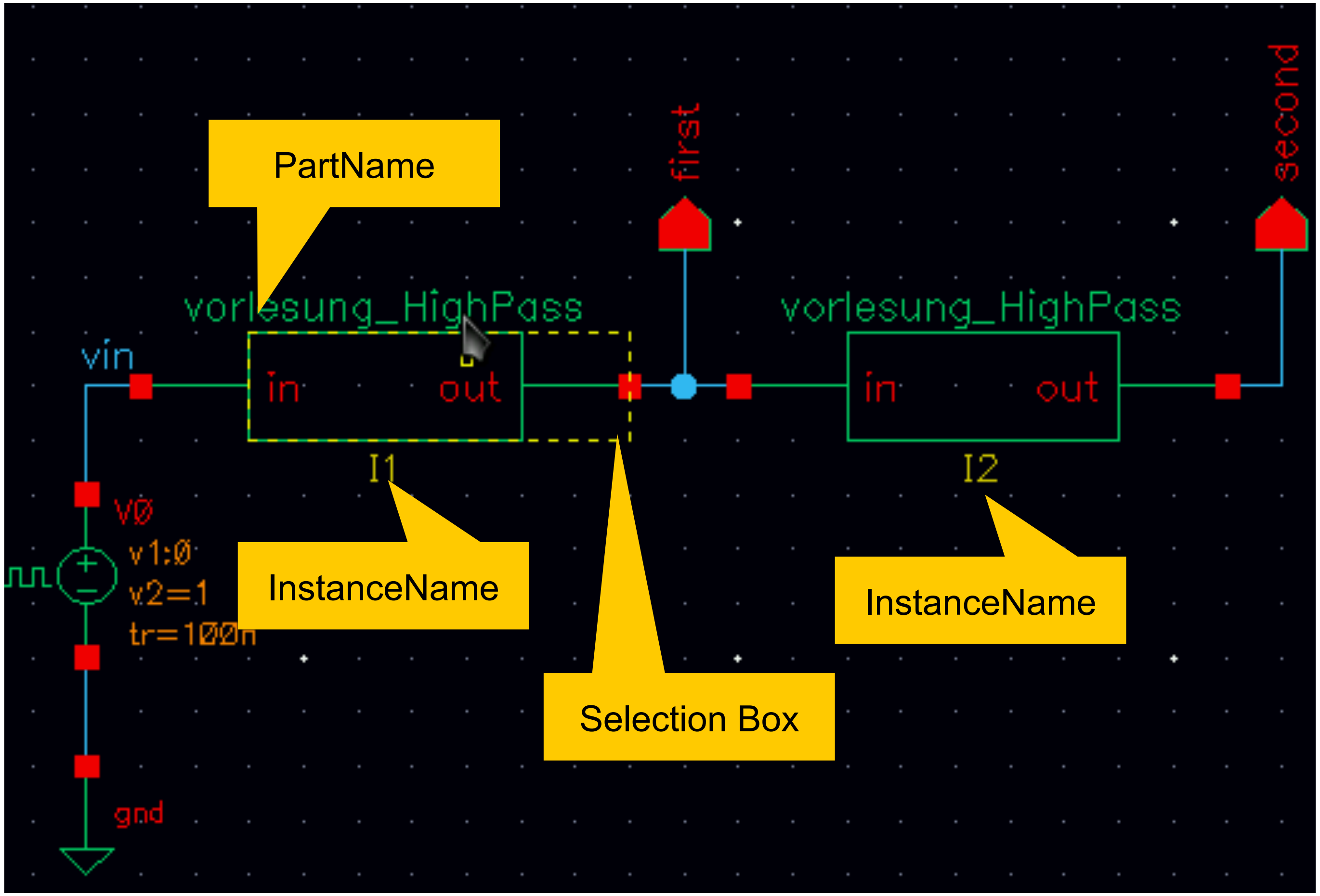
- When you (later) add new pins to the schematic, you **also** have to add them to the symbol.
  - Make sure **name** and **type** are the same!
  - Best copy other pins and rename them

- You can move, stretch, … as usual
- You can change the size or 'justification' of the labels



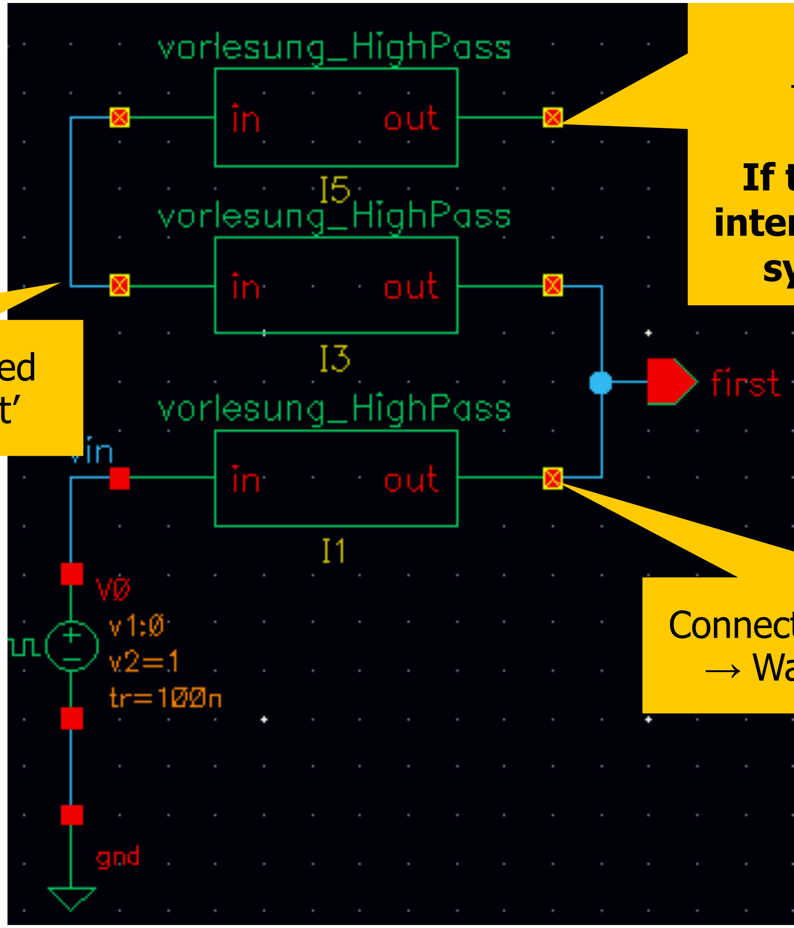
Using the symbol

- In a schematic, you can add your symbol now is the same way as any other instance



Inputs / Outputs / InputsOutputs

- After ‘Check & Save’, warning may pop up in the CIW:



Unconnected **output**
→ Warning ‘floating output’

If this missing connection is intended, connect the ‘noConn’ symbol from library ‘basic’

Leaving **inputs** unconnected
→ Warning ‘floating input’

Connecting several **output** pins
→ Warning ‘shorted output’

- inputOutputs can be connected arbitrarily. Use with caution!

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

TRAVELING THE HIERARCHY

Traveling in the Hierarchy

- Assume you are in Schematic A which contains an Instance of PartType B
- If you want to modify (the symbol or schematic of) B, you normally have to open that cell from the library browser

- You can better *'dive into'* B by
  - Selecting the instance
  - Edit → Hierarchy → Descend Edit (Shift-X)
  - Select the view
  - Select if you want a new window / new tab / use existing tab
- You then end up in symbol / schematic of B
- When done, return back 'up' with Edit → Hierarchy → Return (Shift-B)
- You can also Descend for Read Only (Ctrl-X) or Edit in Place (x). This Edits B but shows A ! Powerful but dangerous!

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

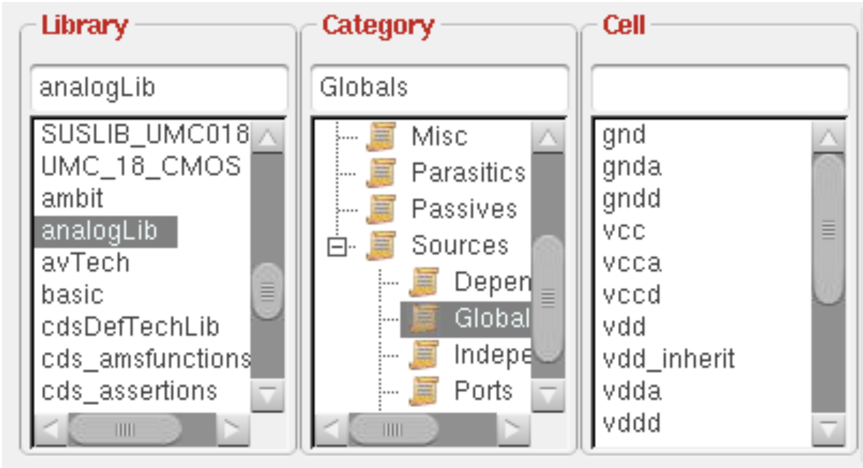
GLOBAL NETS

Global Nets

- A net is normally only known in the open schematic
  - Connecting nets between schematics need pins
- This can be tedious for signals which are used very often
  - analogue / digital power / ground
  - substrate potential
- You can use **global** nets, known **everywhere**
  - They are identified by an **exclamation** mark: xxx**!**
- Typical nets are
  - gnd! or sub! chip substrate
  - gndd! and vddd! digital ground /supply
  - gnda! and vdda! analogue ground / supply
- Handle them with care, because it is hard to track where they are used...

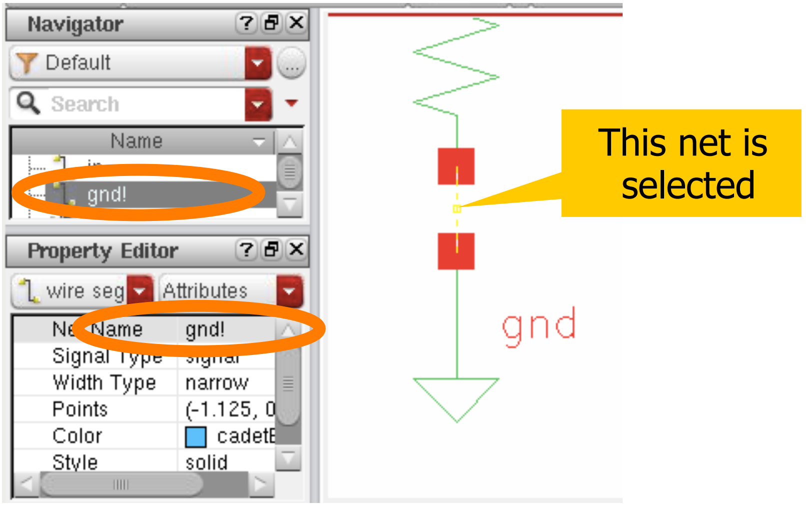
Global Nets

- There are several global 'symbols' in analogLib
  - Under Sources -> Global



- They which connect a net automatically to the corresponding global net

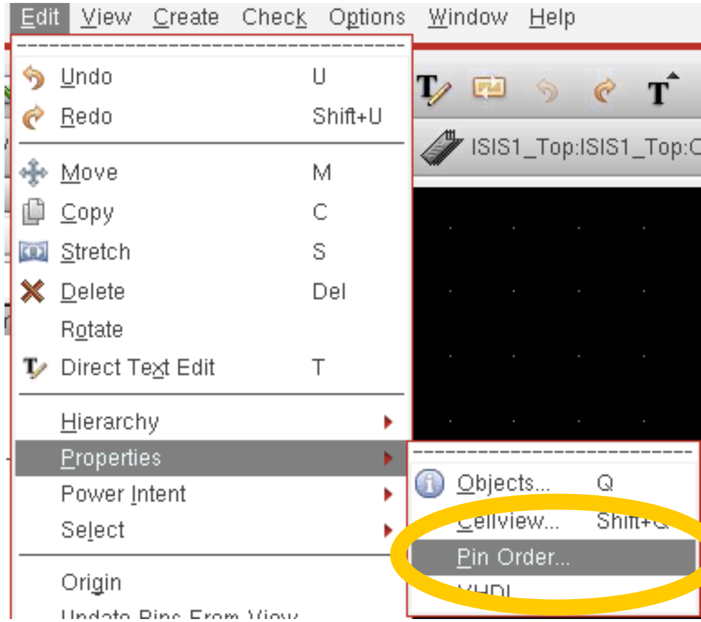
- Connecting to symbol '**gnd**' is the same as labelling a net with '**gnd!**'



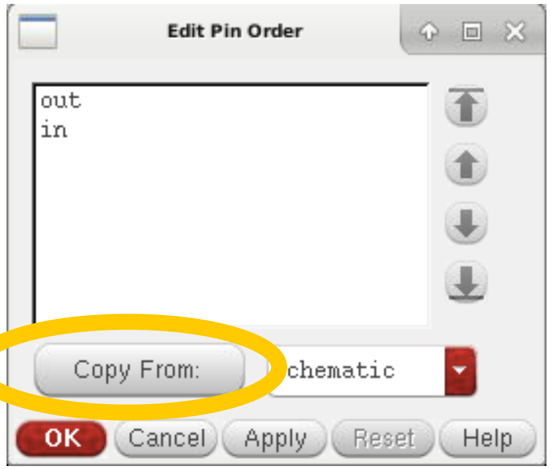
Pin Order

- 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 pins,..

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



- Best copy the Pin Order from another view:



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

BUSSES AND ADVANCED NET NAMING

Advanced Net Names **(Important!)**

- A single 'wire' on the schematic can represent **several** nets, i.e. a 'bus' or bundle of nets.

When a wire has multiple nets assigned:
Imagine the nets **stacked onto each other in the order they are listed**

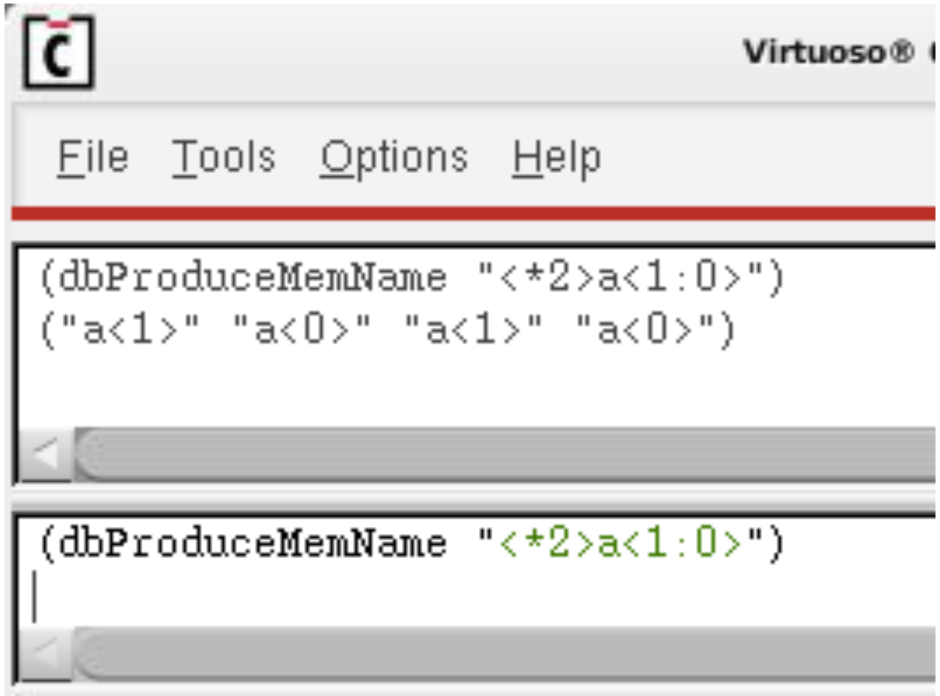
- Examples:
  - simple wire — in
  - Multiple wires — a,b — separated by **comma**
  - Bus — d⟨4:0⟩ — 5 signals: d⟨4⟩,…,d⟨0⟩
  - Bus — x⟨1:5⟩ — different index order: x⟨1⟩,…,x⟨5⟩
  - Repetition — ⟨*3⟩a,⟨*2⟩b — this is equivalent to a,a,a,b,b
  - Skip indices — d⟨7:3:**2**⟩ — = d⟨7⟩,d⟨5⟩,d⟨3⟩
  - Index list — d⟨1:0,3, ⟨*2⟩5⟩ — = d⟨1⟩,d⟨0⟩,d⟨3⟩,d⟨5⟩,d⟨5⟩

- This works for labels and for pins

Advanced Net Names

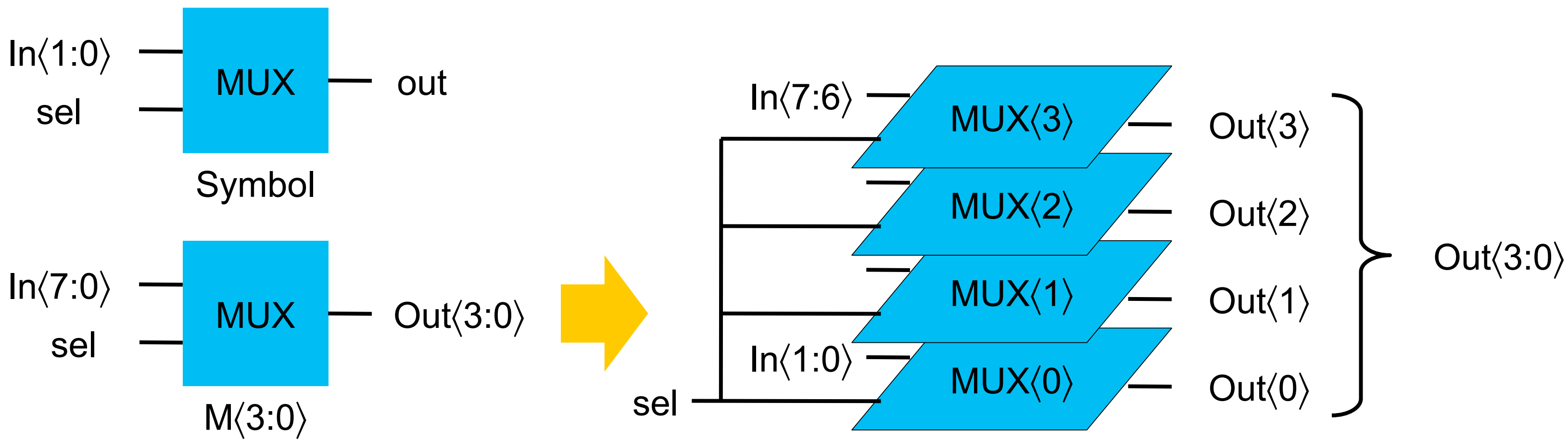
- If you are not certain how a complicated net name expands: Type the expression in the CIW (Command Interpreter Window) using

  (dbProduceMemName "expression")



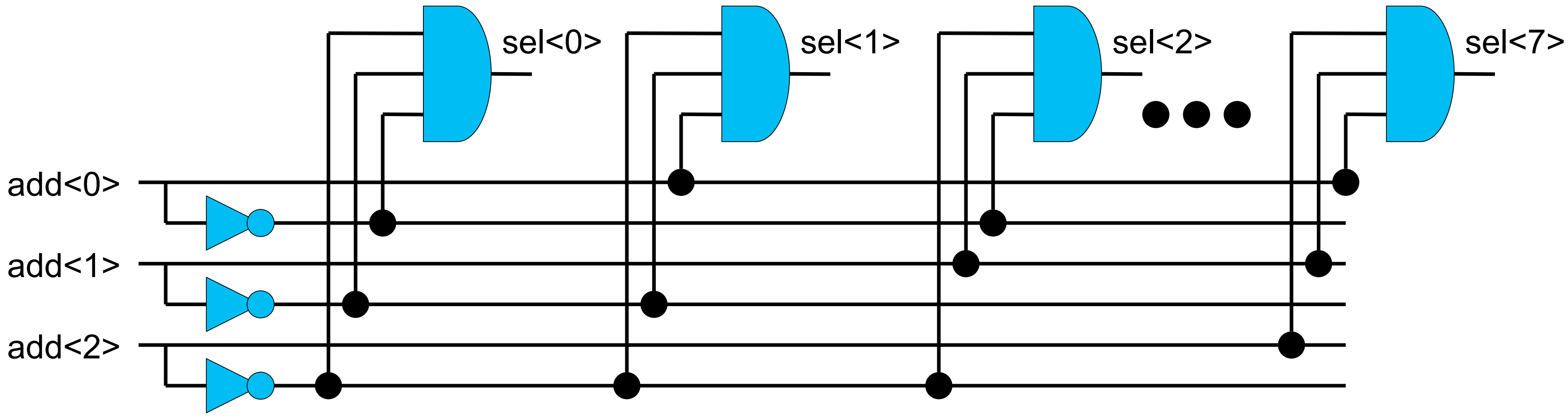
Multiple Symbols **(Important!)**

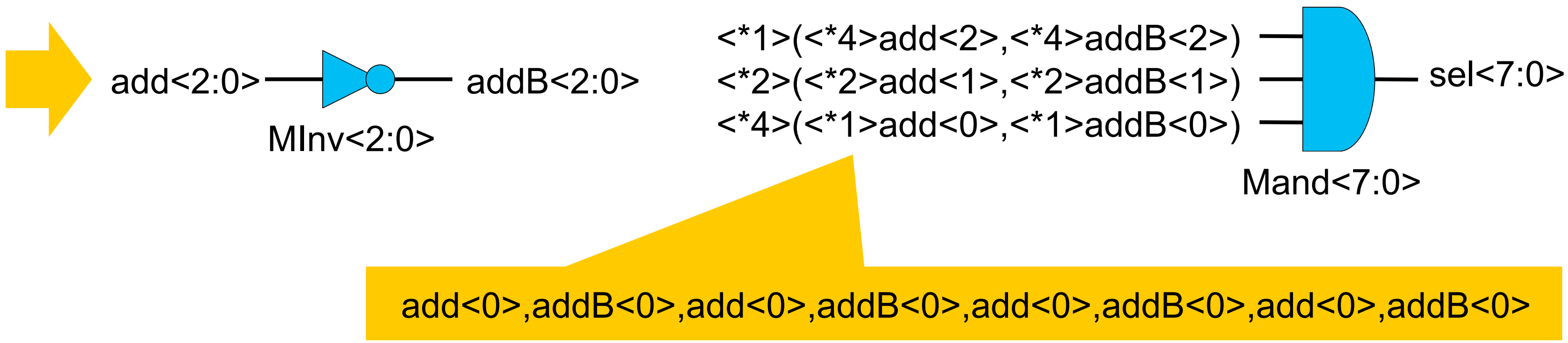
- **Instances** can be indexed as well:
  - An instance with name M⟨3:0⟩ contains 4 elements M⟨3⟩…M⟨0⟩
  - They are (again) lying 'on top of each other' (in the order given)
- The instance **pins** are stacked on top of each other
  - A single pin of N instances becomes a bus which is N nets wide
  - A pin with 2 nets (in⟨1:0⟩) becomes 2N nets wide etc.
  - Connected nets must be have **exact** length OR
    be a **single** wire (see net 'sel' below), connecting **all** nets



Multiple Symbols: 2^{nd} example

- You want to design a 3 Bit address decoder which activates one of 8 output signals sel<7:0> as a function of 3 address inputs add<2:0>:



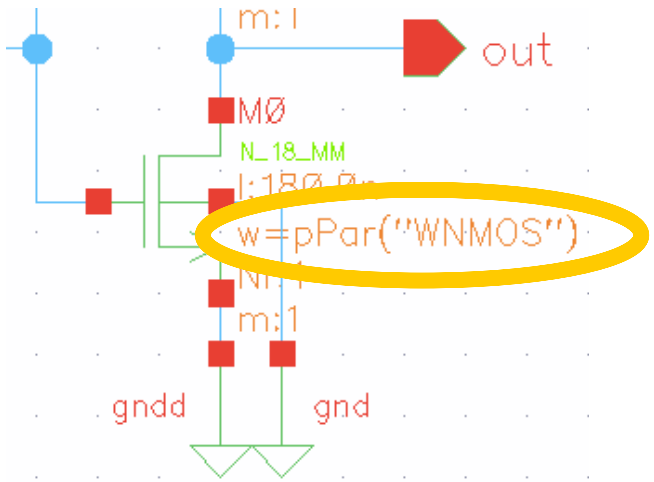


Symbols With Parameters

Parameterized Symbols (Step 1/3)

- It occurs that you need very similar schematics where only a parameter is changes
  - Example: Inverter with different PMOS withs
- Instead of creating multiple views, you can create a view with a PARAMETER:

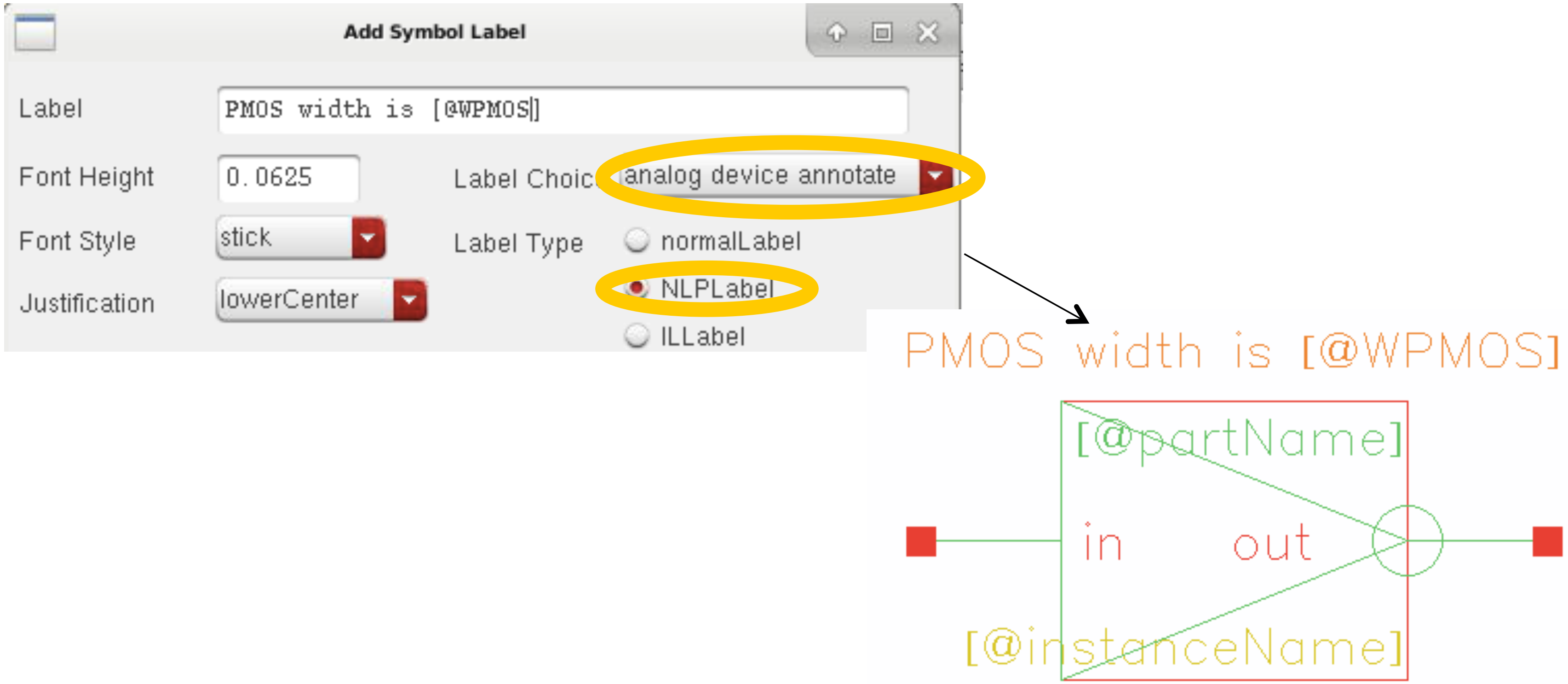
1. In the schematic:
   introduce the parameter with **pPar("name")** (capital P!)



Parameterized Symbols (Step 2/3)

2. In the symbol: Add a label
   - Label Choice: *analog device annotate*
   - Label Type: *NLPLabel*

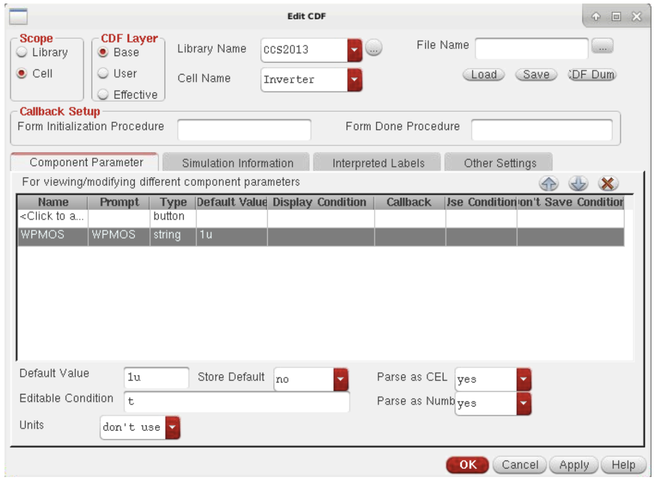
   Add any text, refering to the parameter as [@paramname]



Parameterized Symbols (Step 3/3)

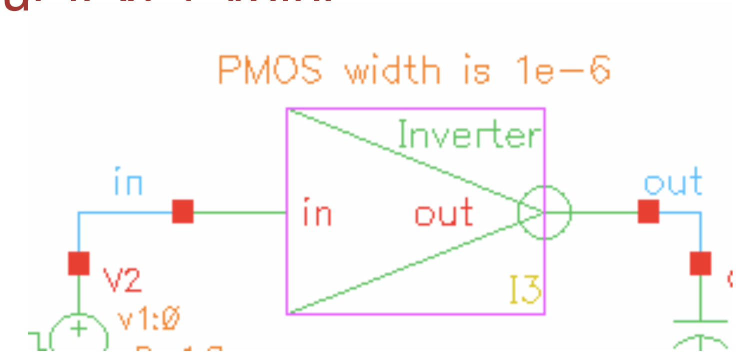
3. Cadence still needs to know about the new parameter:
   - In CIW→Tools→CDF→Edit
   - Choose Scope: Cell
   - Choose CDFLayer: Base
   - Select Cell

   - Add your *paramname* in the form
     - Type: String
     - Set prompt string & default value
     - Store Default: no (=default)
     - Parse as CEL: yes
     - Parse as Number: yes
     - Editable Condition: t (needed ?)
     - Units: don't use (=default)

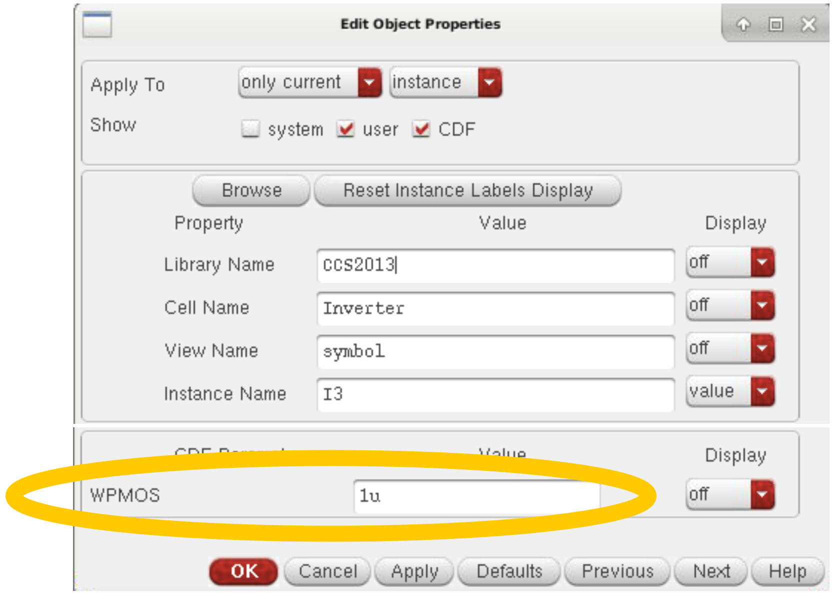


Parameterized Symbols: Instantiation

- The Symbol now shows your text + value
  - You may need to delete and re-instantiate the symbol..



- You can change the parameter in the instance properties



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Inherited Nets

Inherited Nets

- It is possible to over-write nets in schematics (mostly supplies) from a higher hierarchy level.
- This 'inherited nets' approach is not further described here…