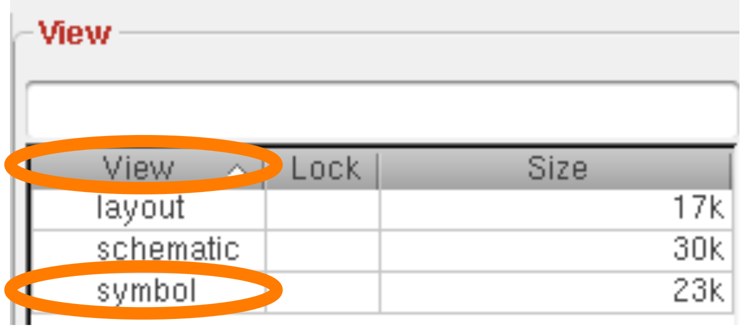
Schematics

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



View

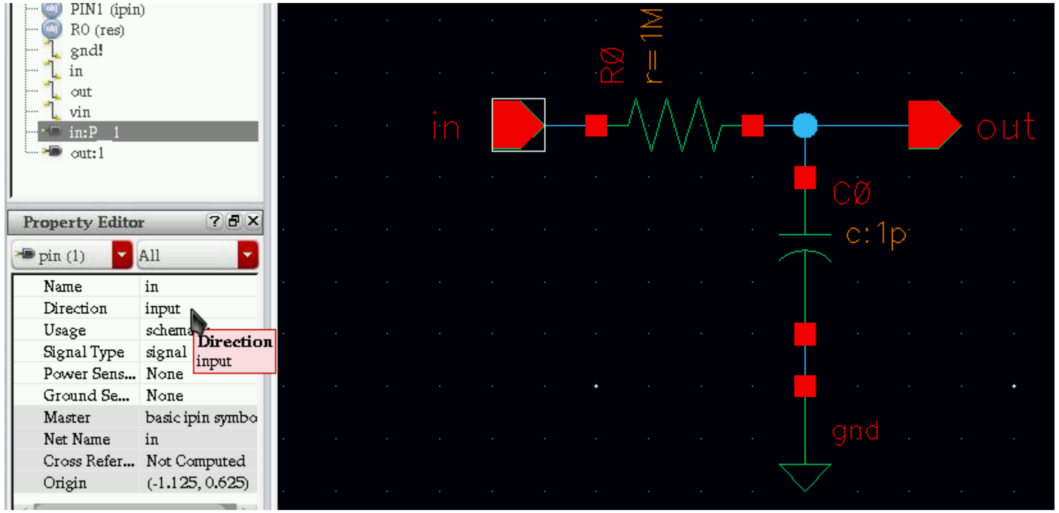
View	Lock	Size
layout		17k
schematic		30k
symbol		23k

- The nets which are 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 must 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 later
  - **InputOutput** are most general. Only use if you have to!



PIN1 (ipin)
R0 (res)
gnd!
in
out
vin
in:P 1
out:1

Property Editor

pin (1) | All

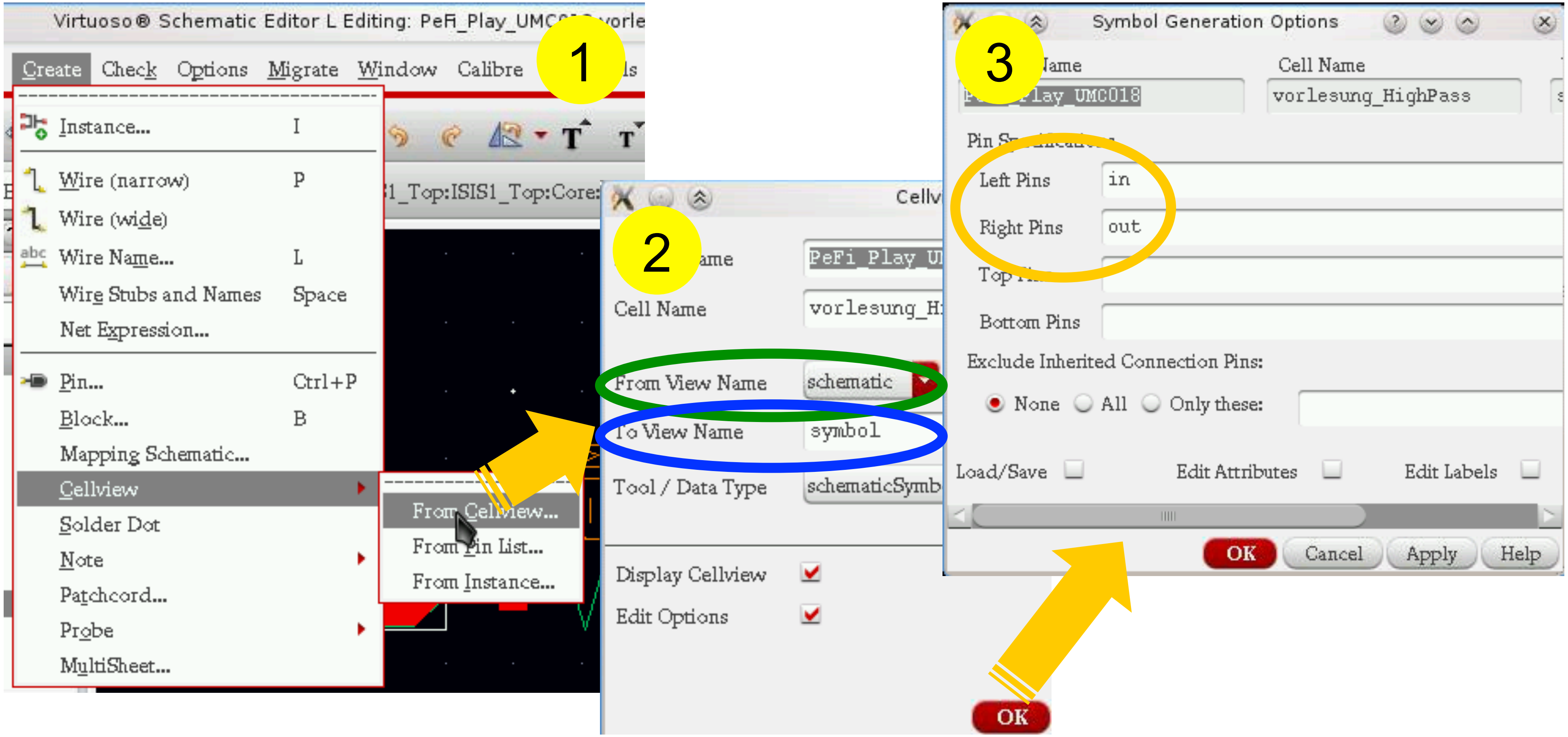
Name	in
Direction	input
Usage	schema
Signal Type	signal
Power Sens...	None
Ground Se...	None
Master	basic ipin symbo
Net Name	in
Cross Refer...	Not Computed
Origin	(-1.125, 0.625)

Direction
input

- A pin **labels** the net, i.e. a further label is not required
- Better remove all symbols used for simulation (sources..)

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 the pin locations



Library Name: PeFi_Play_UMC018

Cell Name: vorlesung_HighPass

From View Name: schematic

To View Name: symbol

Tool / Data Type: schematicSymbol

Display Cellview ☑

Edit Options ☑

Pin Specifications

Left Pins: in

Right Pins: out

Top Pins:

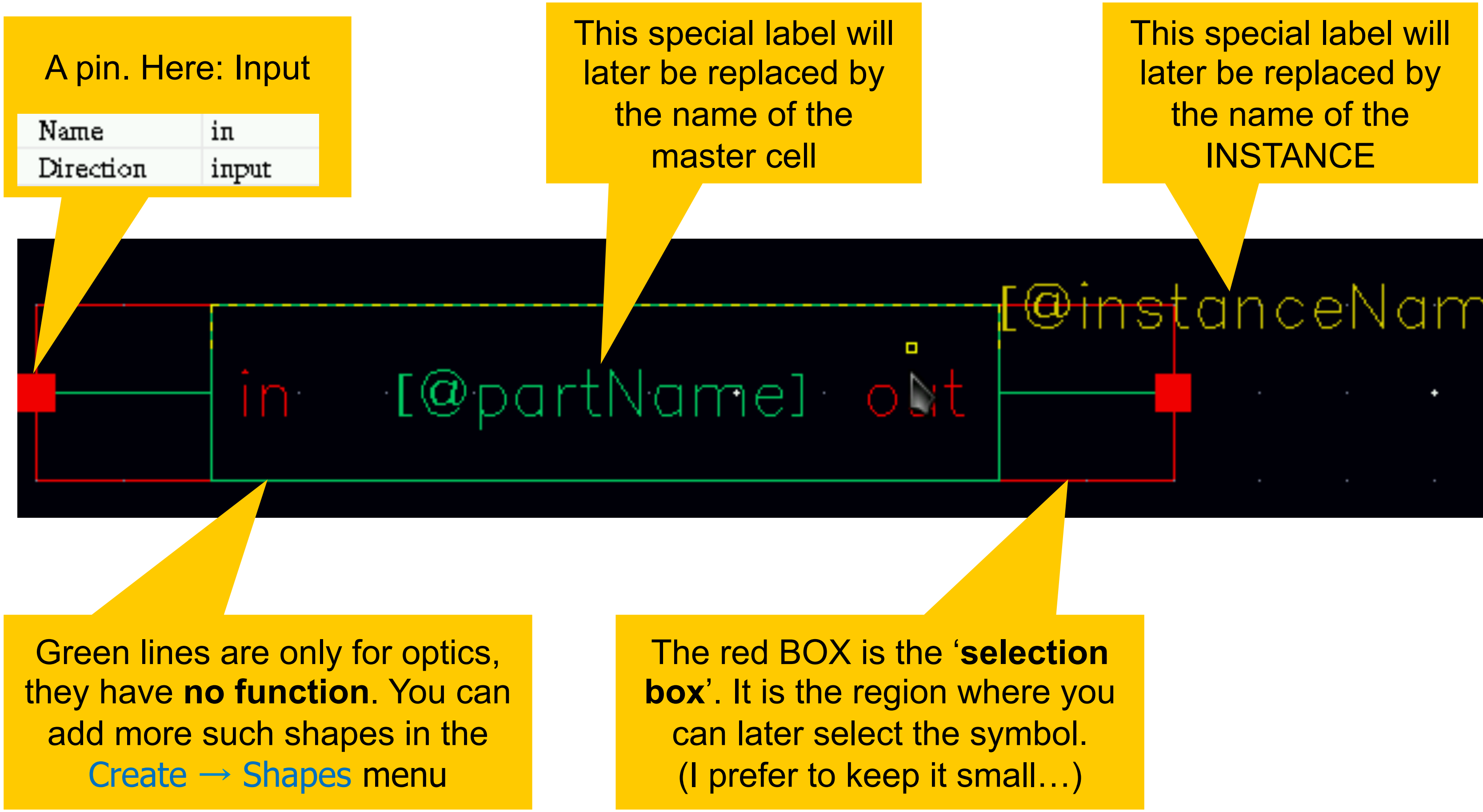
Bottom Pins:

Exclude Inherited Connection Pins: ● None ○ All ○ Only these:

Load/Save □ Edit Attributes □ Edit Labels □

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

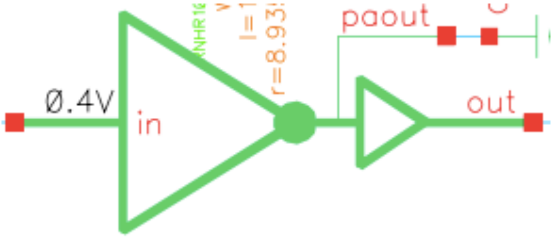
Green lines are only for optics, they have **no function**. You can add more such shapes in the Create → Shapes menu

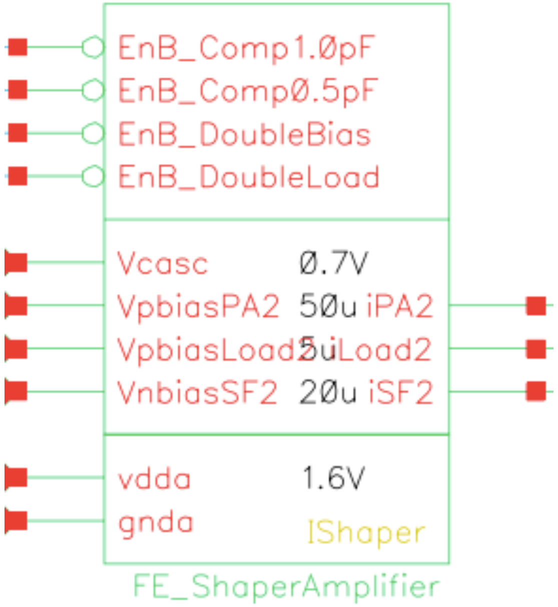
The red BOX is the '**selection box**'. It is the region where you can later select the symbol. (I prefer to keep it small…)

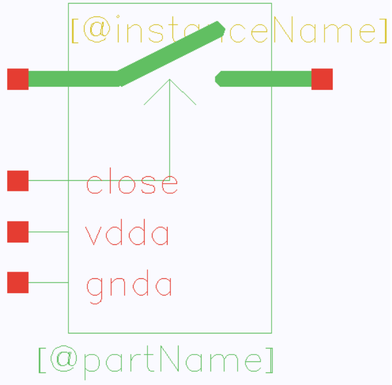
- You can set the origin under Edit → Origin

Make Nice Symbols!

- Your schematics get more readable if the symbols are 'nice':
  - Power (if present at pins) may be grouped at the bottom
  - Group bias signals, use 'good' names
  - Inputs are left / outputs are right
  - Digital signals are grouped
  - Active Low signals have a bullet
  - Clocks have with a triangle
  - Add a little drawing of the functionality Create→Note→Shape or Create→Shape
  - Add text: Create→Note→Text
  - You may delete trivial labels

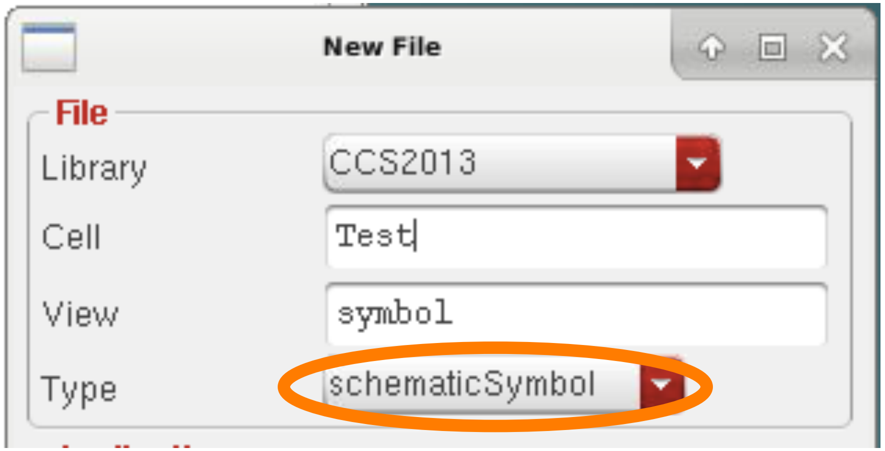






Creating a Symbol from Scratch

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



File

Library: CCS2013

Cell: Test

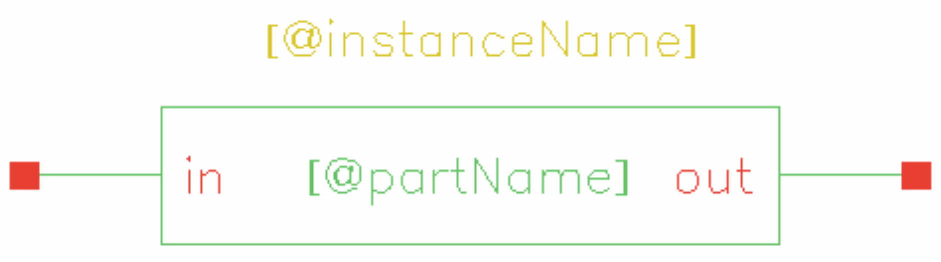
View: symbol

Type: schematicSymbol

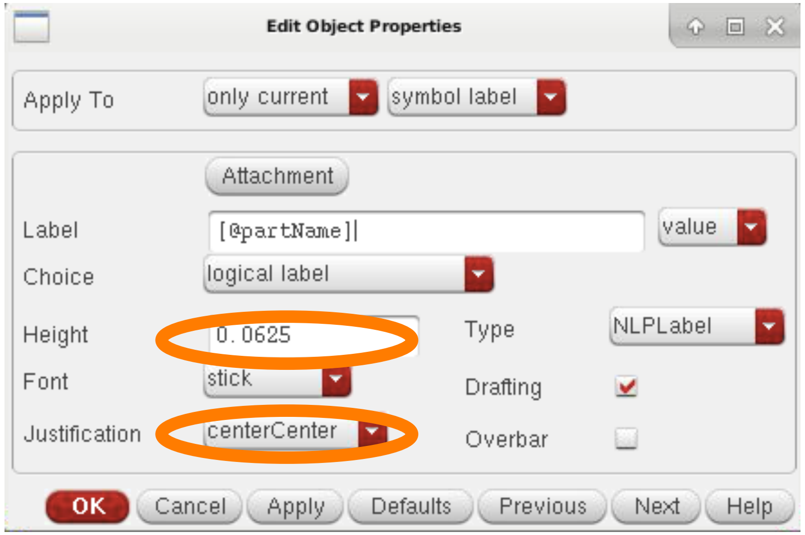
- You must then place all pins, boxes, labels, .. by hand.

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


Edit Object Properties

Apply To: only current, symbol label

Attachment

Label: [@partName] (value)

Choice: logical label

Height: 0.0625

Type: NLPLabel

Font: stick

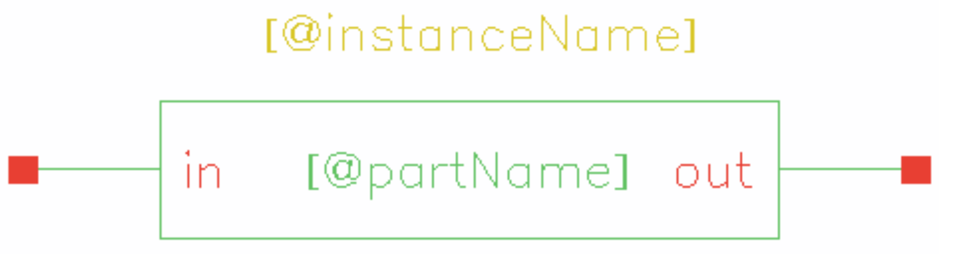
Drafting: ☑

Justification: centerCenter

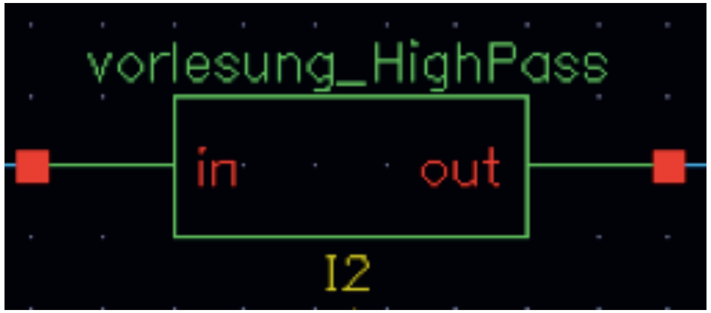
Overbar: □

@InstanceName and @PartName

- Two special labels are created automatically:



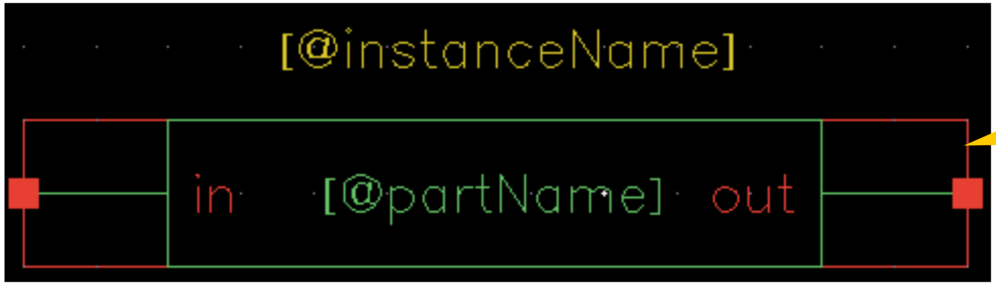
  - [@instanceName] will display – surprise! – the name of the instance (of this symbol) that you place in another schematic, i.e. **I2** or, better, **lamp1** or so
  - [@partName] displays the (library) name of the cell, i.e. **vorlesung_HighPass** or **NAND2**



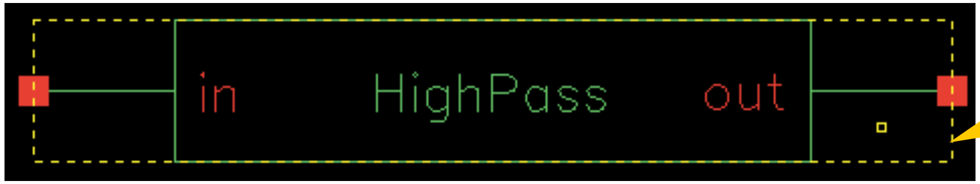
- Place them somehow **nicely** (size / alignment / position)

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***

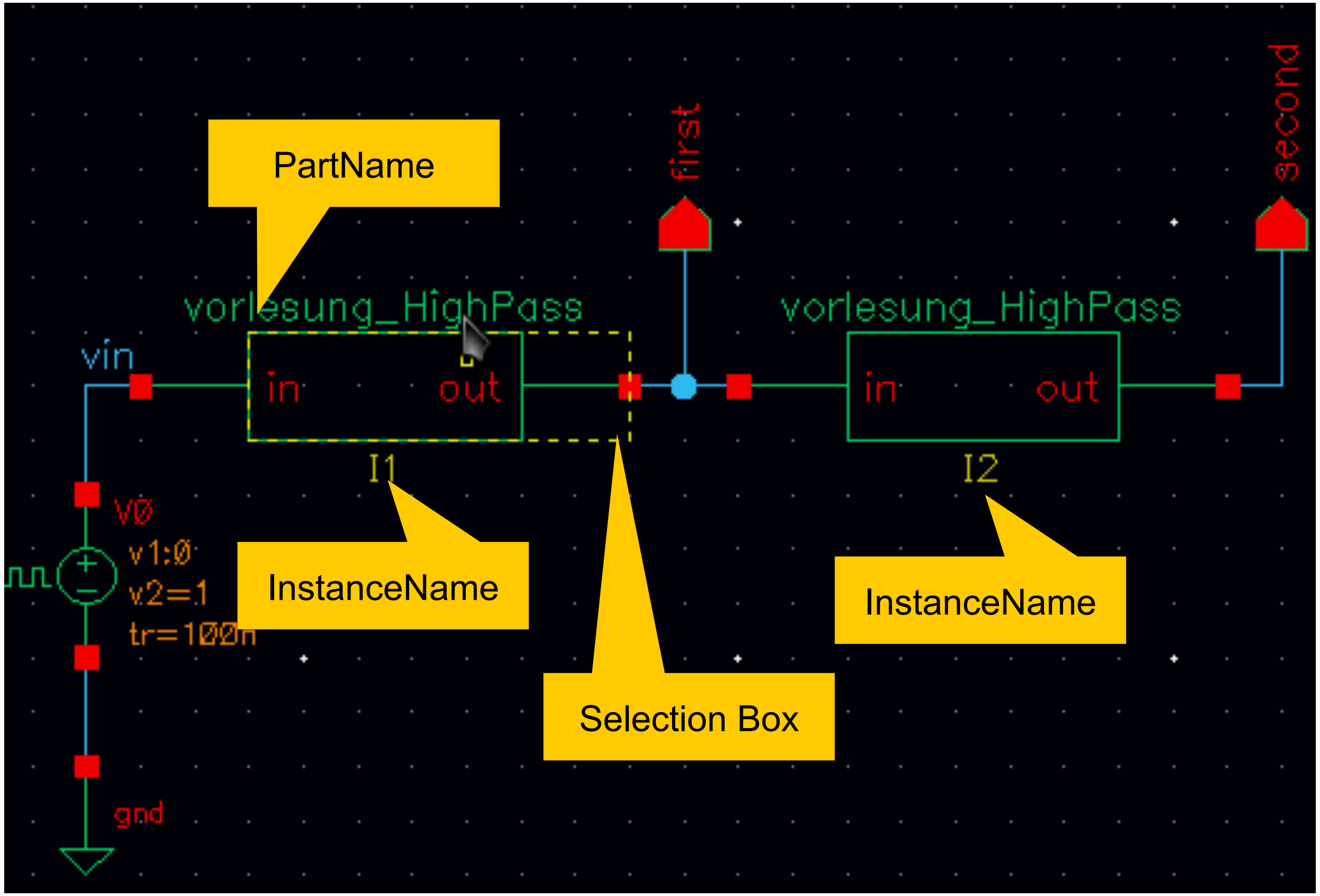


Highlightning in a ***schematic*** which uses an instance of the cell

- 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.

Using the symbol

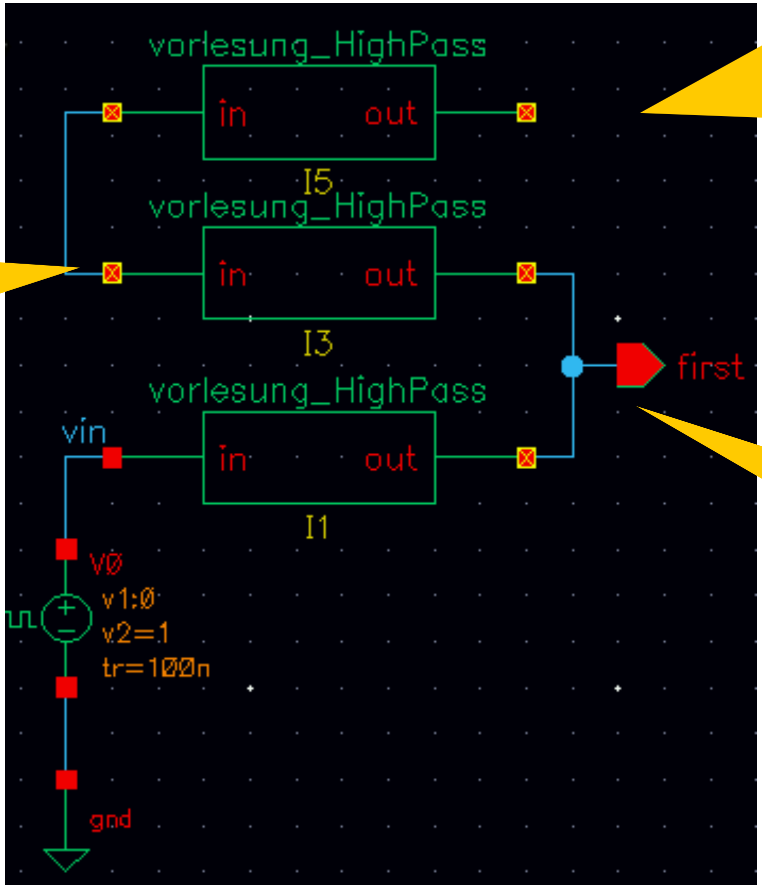
- In a schematic, you can add your symbol now in the same way as any other instance



Inputs / Outputs / InputsOutputs

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

Unconnected inputs
→ Warning ‘floating input’



Unconnected output
→ Warning ‘floating output’

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

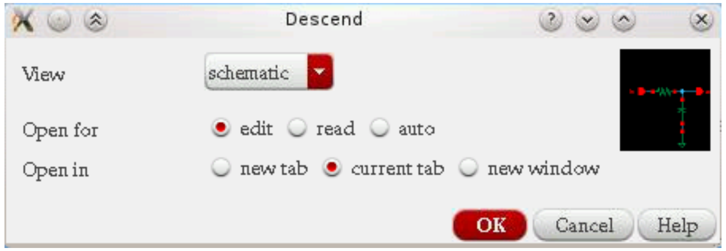
Shorted output pins
→ Warning ‘shorted output’

- InputOutputs can be connected arbitrarily. Use with caution!
- All schematics should be 'clean', i.e. issue no warnings!

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



View: schematic

Open for: ● edit ○ read ○ auto

Open in: ○ new tab ● current tab ○ new window

- 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**!

Global Nets

Global Nets

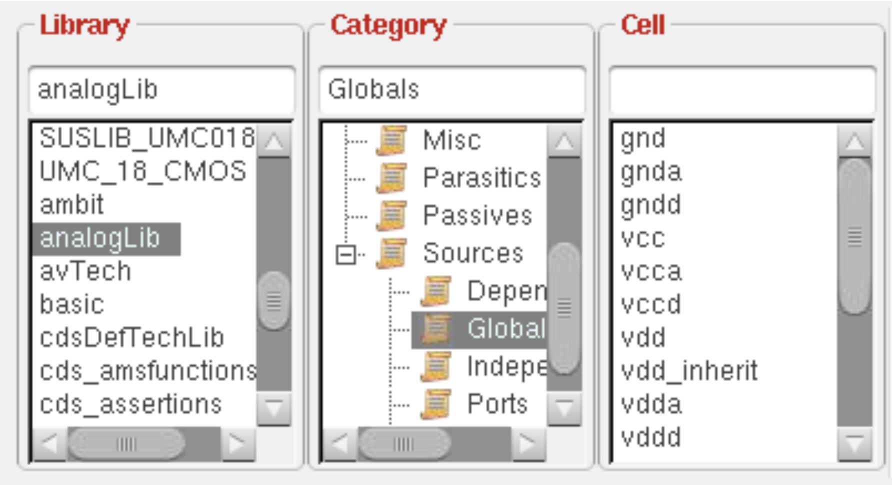
- A net is normally only known in the corresponding schematic
  - Connecting nets *between* schematics requires *pins*
- This can be tedious for signals which are used very often
  - analogue / digital power / ground
  - substrate potential
- You can use **global** nets, known **everywhere**
  - They are identified by an **exclamation** mark**:** xxx**!**
- Common global 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



Library: analogLib

SUSLIB_UMC018
UMC_18_CMOS
ambit
analogLib
avTech
basic
cdsDefTechLib
cds_amsfunctions
cds_assertions

Category: Globals

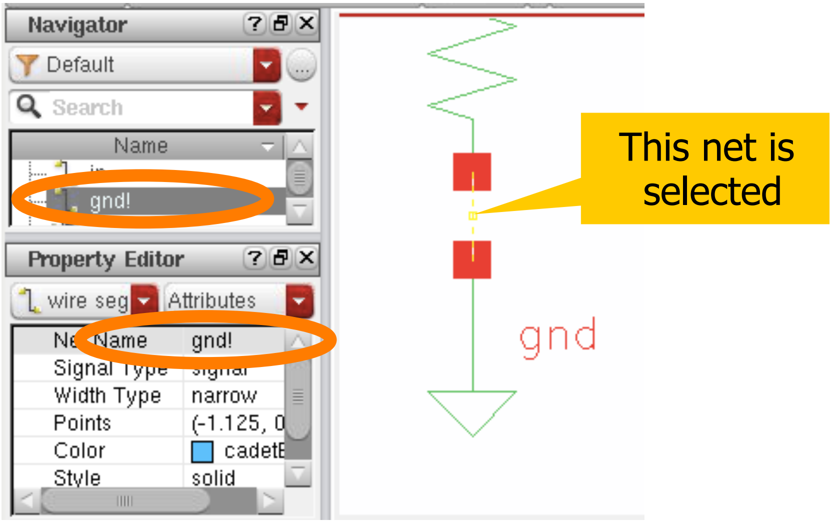
Misc
Parasitics
Passives
Sources
Depen
Global
Indepe
Ports

Cell

gnd
gnda
gndd
vcc
vcca
vccd
vdd
vdd_inherit
vdda
vddd

- They connect a net automatically to the corresponding global net

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



Navigator
Default
Search
Name
gnd!

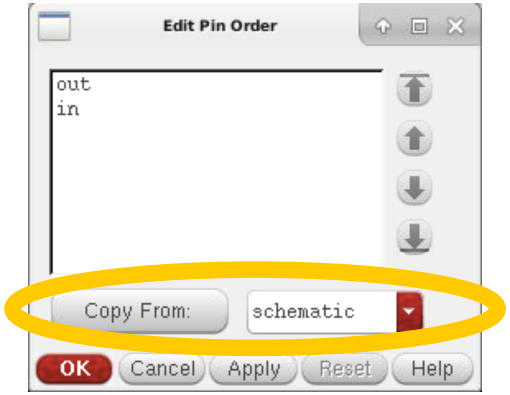
Property Editor
wire seg | Attributes

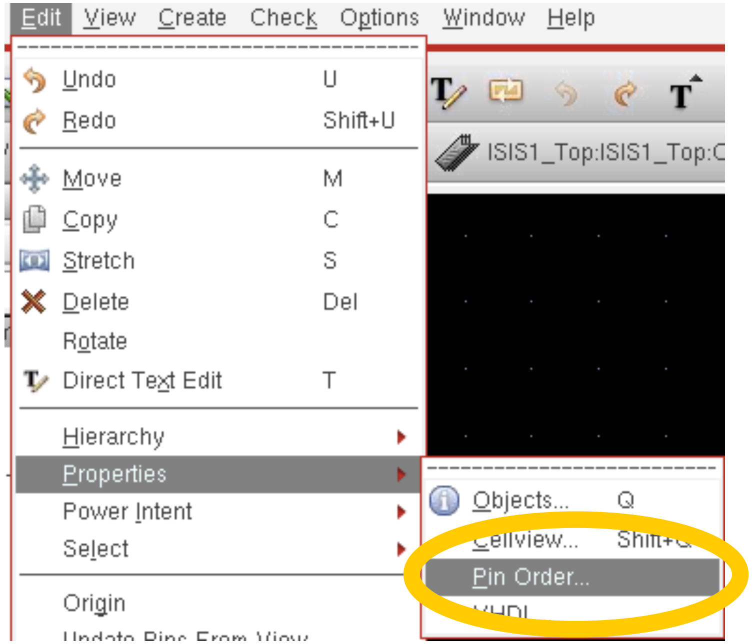
Net Name	gnd!
Signal Type	signal
Width Type	narrow
Points	(-1.125, 0
Color	🟦 cadetB
Style	solid

This net is selected

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:





- In rare cases, you have to regenerate (for instance) the symbol. (There is a step which allows you to just 'repair' the wrong stuff so that you nice drawing is not affected)

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:

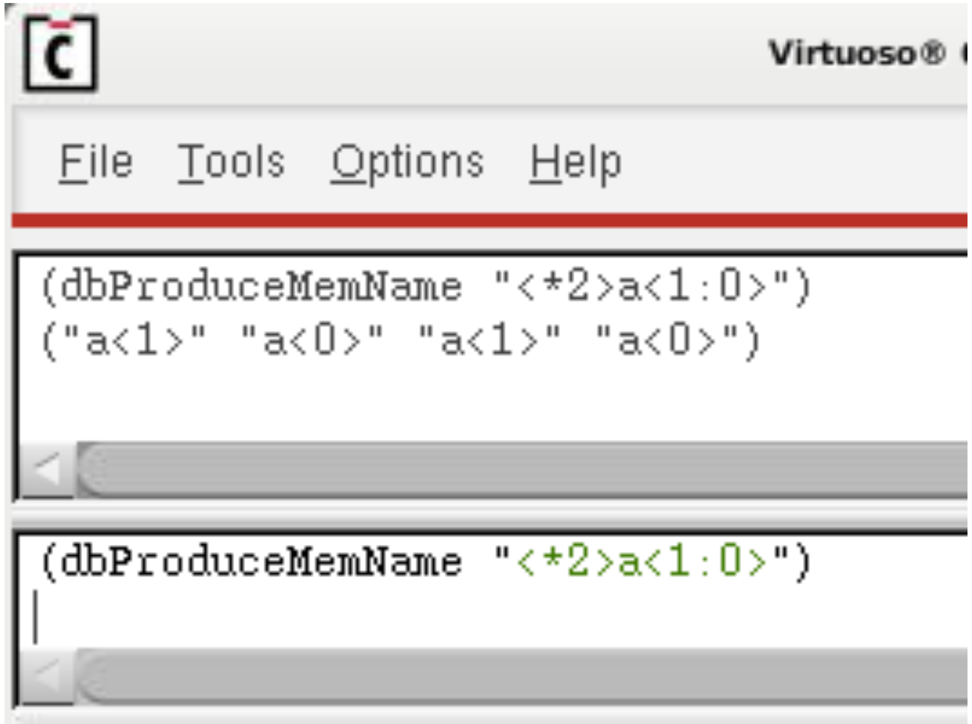
• simple wire	in	
• Multiple wires	a,b	separated by **comma**
• Bus	d⟨4:0⟩	5 signals: d⟨4⟩,…,d⟨0⟩
• Bus	x⟨1:5⟩	different index order: x⟨1⟩,…,x⟨5⟩
• Repetition	⟨*3⟩a,⟨*2⟩b	this is equivalent to a,a,a,b,b
• Skip indices	d⟨7:3:**2**⟩	= d⟨7⟩,d⟨5⟩,d⟨3⟩
• Index list	d⟨1:0,3, ⟨*2⟩5⟩	= d⟨1⟩,d⟨0⟩,d⟨3⟩,d⟨5⟩,d⟨5⟩

- This works for labels and for pins (but use only busses!)

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")

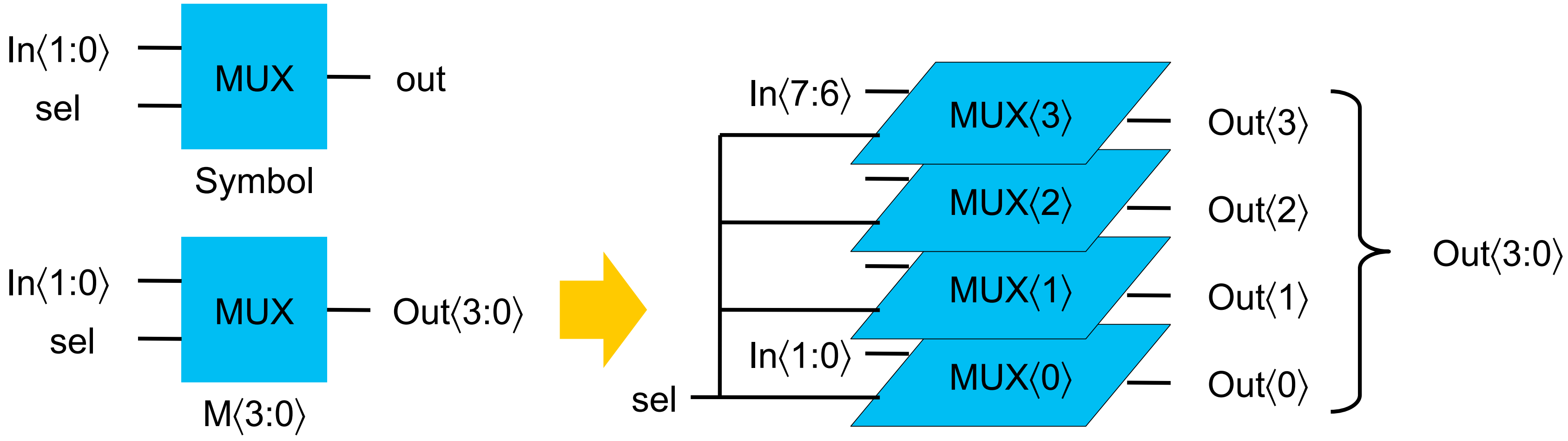


```
(dbProduceMemName "<*2>a<1:0>")
("a<1>" "a<0>" "a<1>" "a<0>")
```

```
(dbProduceMemName "<*2>a<1:0>")
```

Multiple Symbols **(Important!)**

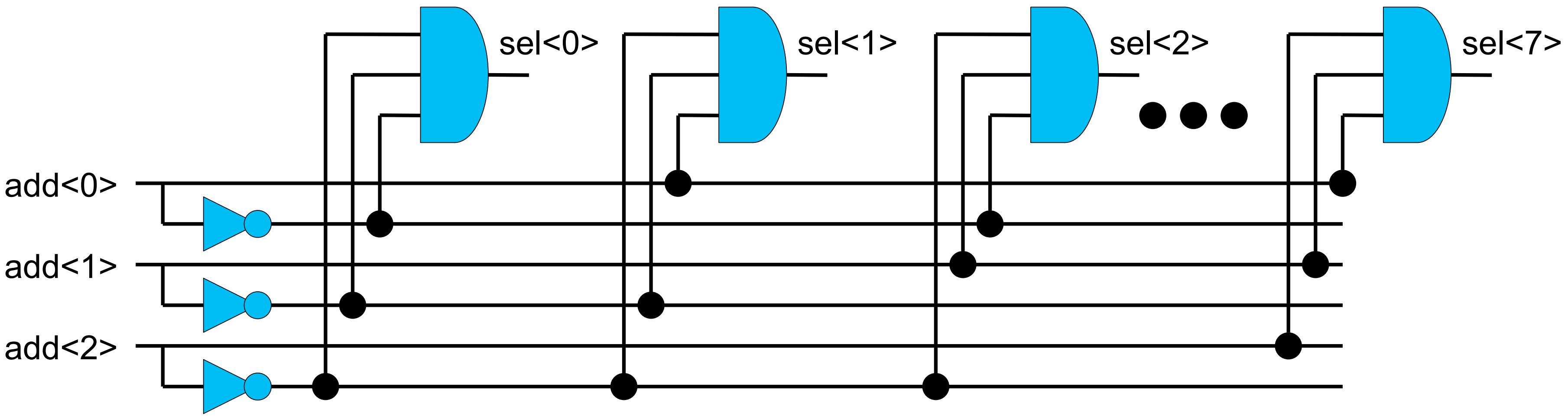
- **Instances** can be indexed as well:
  - An instance with name M⟨3:0⟩ contains 4 elements M⟨3⟩…M⟨0⟩
  - They are (again) lying 'on top of each other' (in the order given)
- The instance **pins** are stacked on top of each other
  - A single pin of N instances becomes a bus which is N nets wide
  - 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



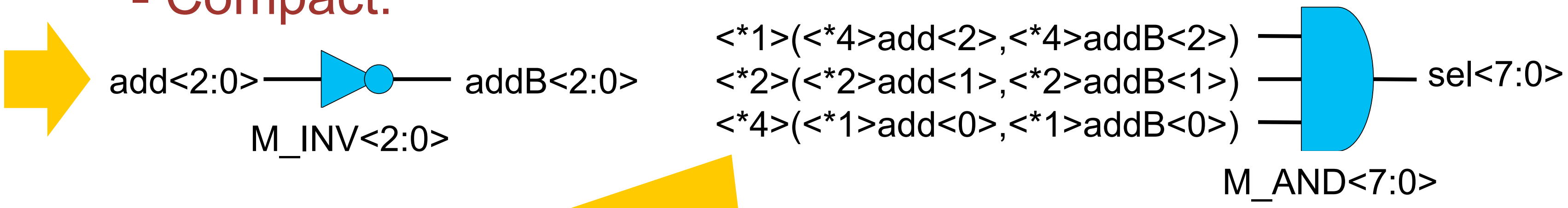
RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Multiple Symbols: 2nd example

- Here is 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⟩:



- Compact:


add<2:0>
addB<2:0>
M_INV<2:0>
<*1>(<*4>add<2>,<*4>addB<2>)
<*2>(<*2>add<1>,<*2>addB<1>)
<*4>(<*1>add<0>,<*1>addB<0>)
sel<7:0>
M_AND<7:0>

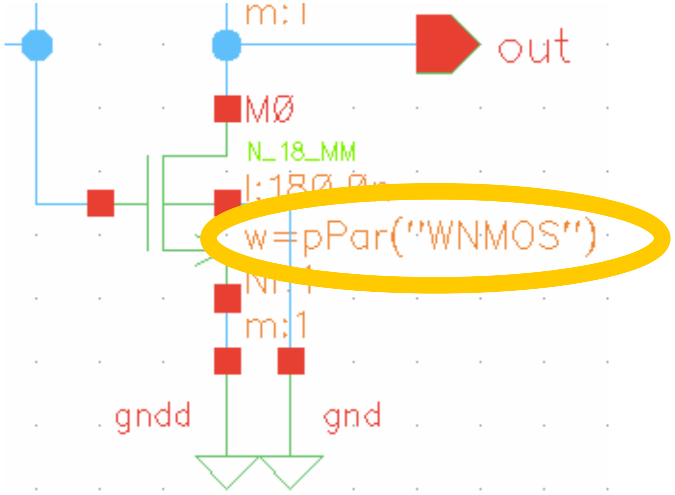
This is: add<0>,addB<0>,add<0>,addB<0>,add<0>,addB<0>,add<0>,addB<0>

Symbols With Parameters

Parameterized Symbols (Step 1/3)

- It occurs that you need very similar schematics where only few parameters are changed (often transistor sizes)
  - Example: Inverter with different PMOS widths
  - Unfortunately, parameters cannot be used everywhere…
- Instead of creating multiple cells, you can create a view with a PARAMETER:

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

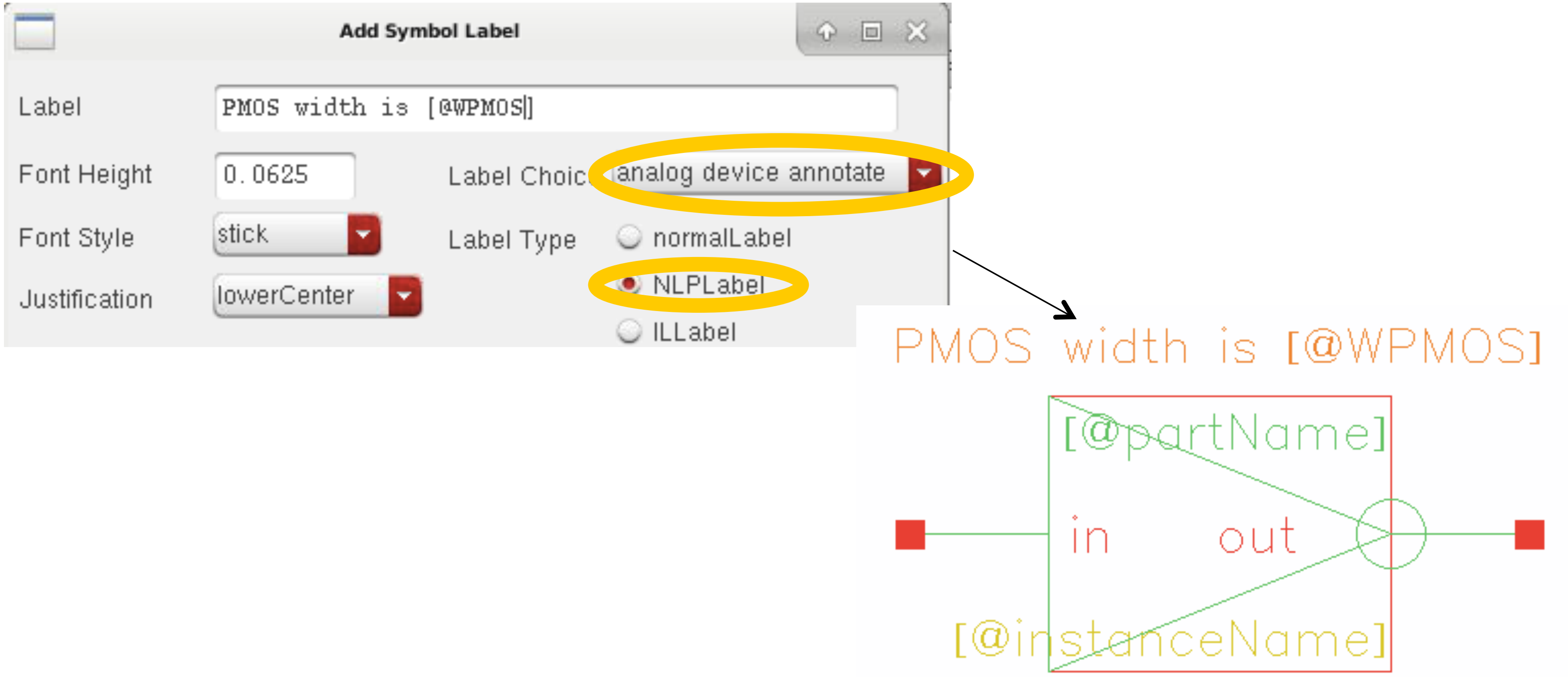


RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Parameterized Symbols (Step 2/3)

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

   Add any text, referring to the parameter as [@pname]



Add Symbol Label

Label: PMOS width is [@WPMOS]

Font Height: 0.0625

Label Choice: analog device annotate

Font Style: stick

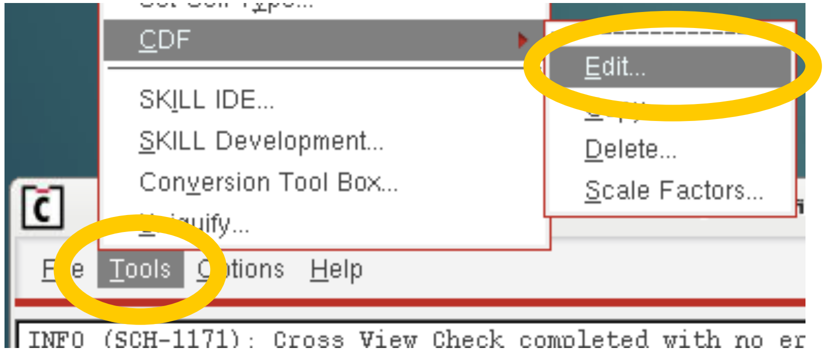
Label Type: ○ normalLabel ● NLPLabel ○ ILLabel

Justification: lowerCenter

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

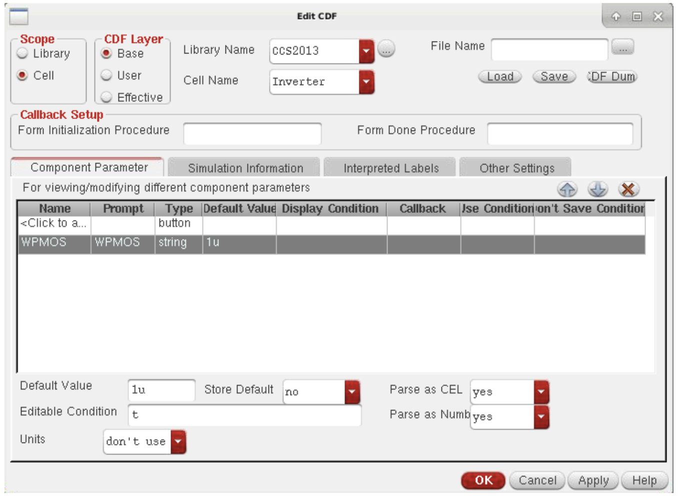
Parameterized Symbols (Step 3/3)

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



```
INFO (SCH-1171): Cross View Check completed with no er
```

   - Add your *pname* 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)


Edit CDF

Scope: ○ Library ● Cell

CDF Layer: ● Base ○ User ○ Effective

Library Name: CCS2013

Cell Name: Inverter

File Name:

Callback Setup

Form Initialization Procedure:

Form Done Procedure:

For viewing/modifying different component parameters

Name	Prompt	Type	Default Value	Display Condition	Callback	Jse Condition	on't Save Condition
<Click to a...		button					
WPMOS	WPMOS	string	1u				

Default Value: 1u

Store Default: no

Parse as CEL: yes

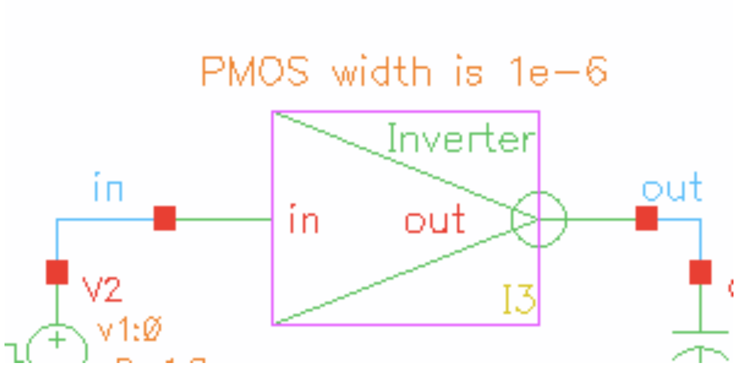
Editable Condition: t

Parse as Numb: yes

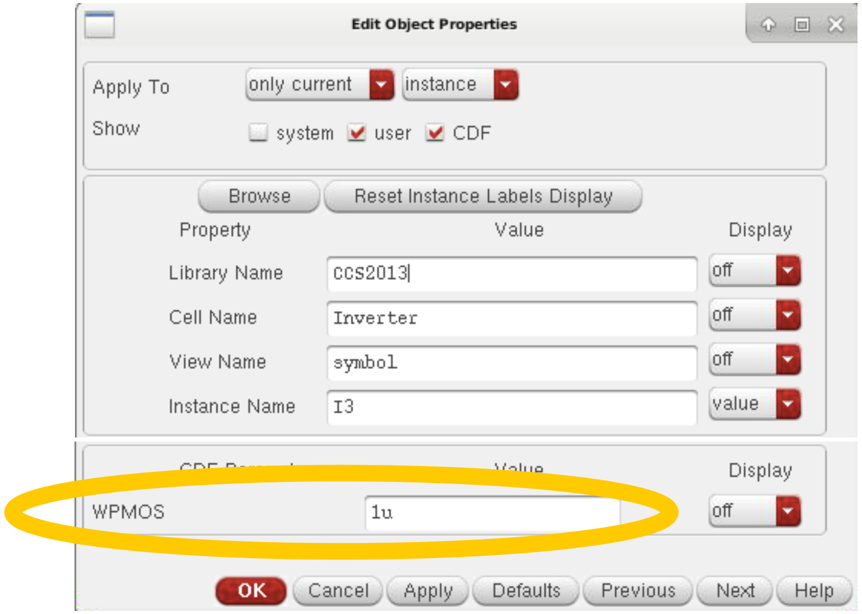
Units: don't use

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



Edit Object Properties

Apply To: only current, instance

Show: □ system ☑ user ☑ CDF

Property	Value	Display
Library Name	ccs2013	off
Cell Name	Inverter	off
View Name	symbol	off
Instance Name	I3	value

CDF Parameter	Value	Display
WPMOS	1u	off

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…

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Features of the Schematic Editor

Some Features

- The additional display of net names at the pins of transistors is often confusing.
  - You can turn this off under View → Hide Terminal Labels

- If you want to see all places where a net in one schematic connects:
  - Enable Highlighting under View → Enable Dynamic Highlighting

- If you want to follow a signal through the hierarchy:
  - Highlight the net under Create → Probe → Add Net