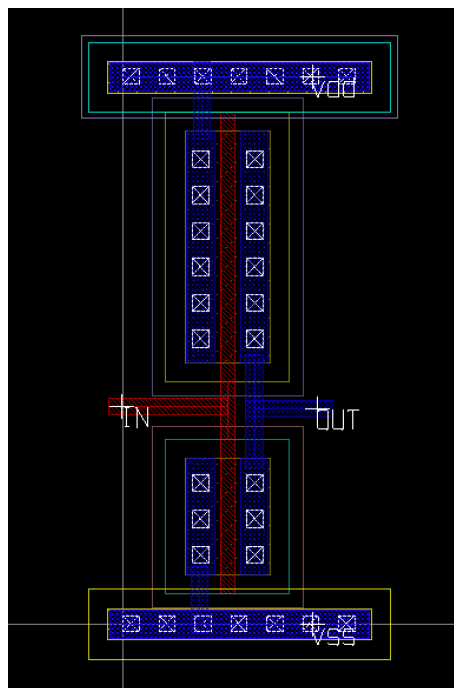


Glade training part 3 - lab

- Start in the training directory
 - `cd glade_training/Part3`
- Make sure your PYTHONPATH includes the location of the training directory Part3:
 - `PYTHONPATH=.:glade_training/Part3:$GLADE_HOME:$PYTHONPATH`
- Open Glade from the cmd line or icon
- Open the library 'part3'
- Open the cellView from library part3 'inv' 'layout':



- Verification->DRC->Run... and choose the Part3/drc.py rules file
 - Browse through the errors using Verification->DRC->View Errors...
 - You can try and fix the errors!

- Next, run extraction.
 - Verification->Extract->Run LPE...
 - Choose the Part3/extract.py rules file
 - View the resulting extracted view. Try selecting shapes and see net info.
- Now run LVS.
 - Verify->LVS->Run LVS...
 - Choose the netlist 'inv.cdl'
 - Note a summary file 'inv.lvs' is written to the working directory.
- Let's try some Python. First, open the 'inv' 'layout' cellView.
- In the command line, type:
 - `cv = getEditCellView()`
- Let's get the cellView's bounding box, and show it:
 - `box = cv.bBox()`
- We can show the coordinates of the box using `left()`, `right()`, `top()`, `bottom()`:
 - `box.left()`
 - This will print the left coord of the bounding box in database units (1/1000th's of a micron)
- Now let's print the instance names in this cellView In the commands below, <tab> means use the Tab key to indent the line, as Python requires indentation for the body of e.g. for loops :
 - `insts = cv.getInsts()`
 - `for i in insts :`
 - <tab> `print i.instName()`
 - <newline with no tab>
- Try finding the metal1 shapes in the cellView:
 - `lpp = cv.getLpp('metal1','drawing')`

- `shapes = cv.getShapes(lpp)`
- `for shape in shapes :`
- `<tab>print(shape.objType())`
- `<newline with no tab>`

●

This concludes the lab session for Part 3.