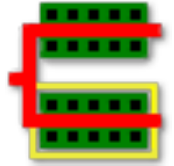


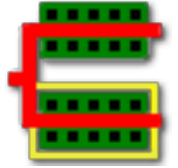
# Glade

Peardrop Design Systems  
29<sup>th</sup> June 2021

# Agenda

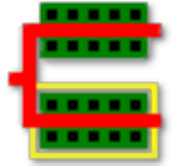


1. Creating objects
2. Selection
3. Editing
4. Boolean operations
5. PCells



# 1. Creating Objects

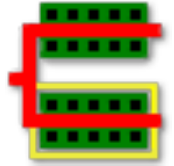
- Shapes
  - Rectangle
  - Polygon
  - Path
  - Label
  - MultiPartPath
- Instance / Array
- Via
- Group



# Creating Shapes

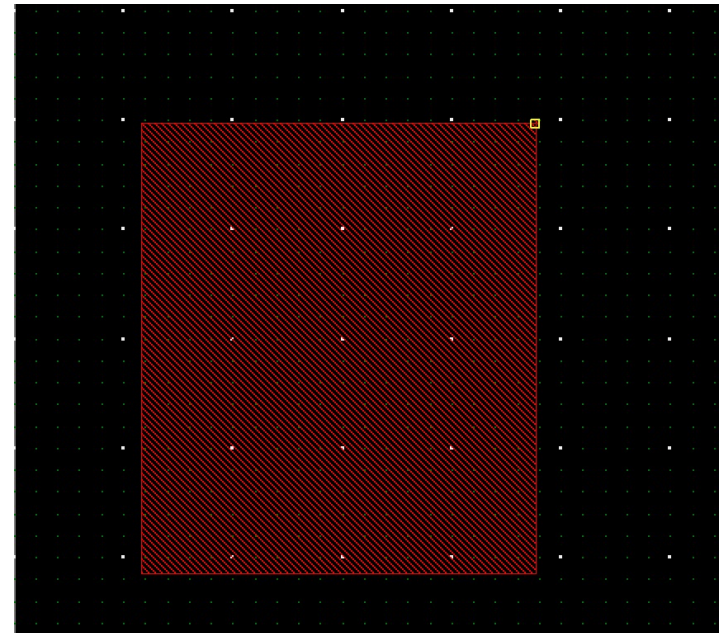
- Select the layer you want to draw on by LMB clicking on it in the LSW
- Use either menu, toolbar or shortcut key to create objects. Shortcut key has the advantage of being able to work in infix mode.
  - Infix mode uses current cursor position for the first coordinate. Set using Display Options form.
- During shape creation an options form can be displayed. It can be toggled between shown/hidden using the F3 key.
- For paths and polygons, finish by hitting Enter key, or by double clicking. There is no need to 'close' a polygon.



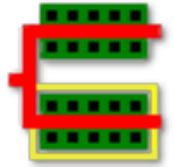


# Create Rectangle

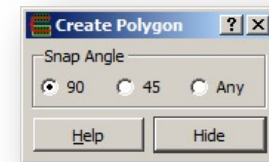
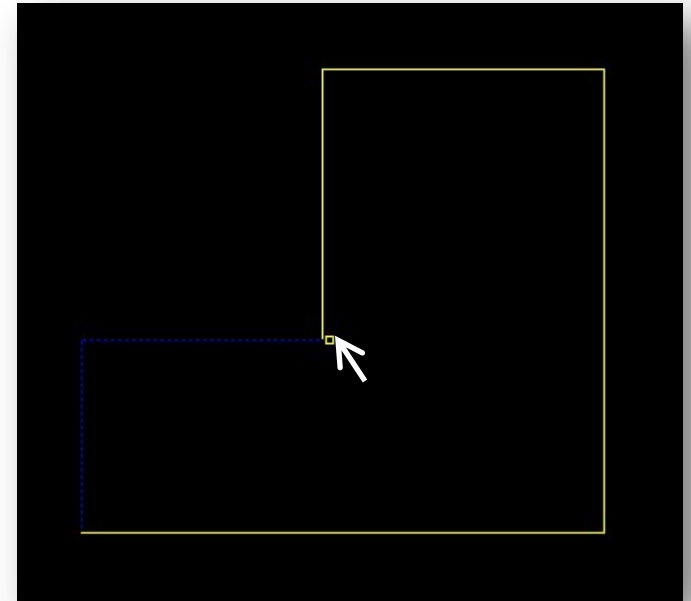
- A rectangle has the following attributes:
  - Origin (x,y)
  - Width
  - Height
  - Layer
- Bindkey 'r' starts a rectangle
  - Or menu, toolbar icon
- Enter 2 coordinates (opposite corners)
- No option dialog for creating rectangles



# Create Polygon

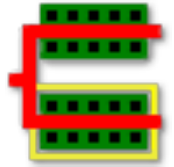


- Bindkey Shift+P starts a polygon
- Edges entered are drawn in yellow
- 'Close edges' are drawn dotted blue. These are edges that will be automatically added to close the polygon, if Enter is pressed or a double click at the current point.
- There is no need to enter the last point coincident with the start point as a result.



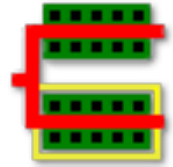
Create Polygon options dialog

# Create Polygon – continued




- Polygon points entered are snapped to the snap grid (defined in the Display Options form) and according to the snap angle (Manhattan / Diagonal / Any Angle).
- While entering a polygon, you can undo the last point entered by using the Backspace key.
- Colinear points entered (points with exactly the same X and Y values) are automatically removed.
- Self-intersecting polygons are not allowed. An error will be given and the polygon will not be created.

# Create Polygon – continued



Properties for object POLYGON in cell test2 view layout

Properties Polygon

Layer  active dwg

Num Vertices 5

Vertices ((8.995,3.090) (16.225,3.090) (16.225,10.255) (13.405,10.255) (8.995,5.845) )

Area 42.0789 Perimeter 26.206

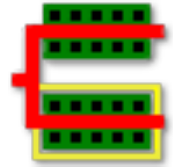
Bounding box (8.995 3.090) (16.225 10.255)

☐ Change all selected objects

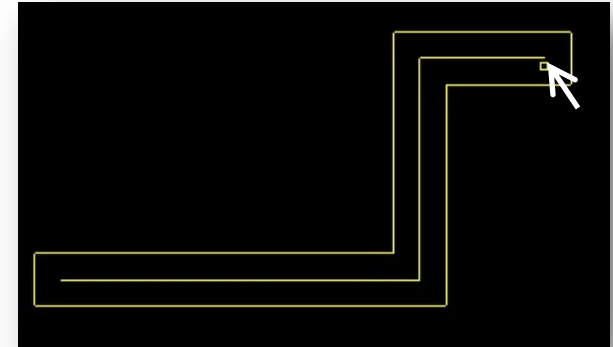
< Previous Next > Delete Apply OK Cancel

- Polygons can be selected and queried to make modifications
  - Change the layer
  - Edit vertices textually
- Query form shows other useful info like area, perimeter, bounding box.

# Create Path



- Create Path
  - Creates a path. Width, extension, style and shielding can be defined in the options form
  - With layer function and vias defined in techfile, the U key will add a via and path creation continues on the next routing layer up. The D key will add a via and path creation continues on the next routing layer down.
- End a path with Enter or double click. Backspace to undo last point entered.



Create Path

Snap Angle

☐ HV ☐ VH ☒ 90 ☐ 45 ☐ Any ☐ Horiz ☐ Vert

Width 0.18 ☐ Keep width constant Style EXTEND

Begin Ext 0.09 End Ext 0.09

Path Net

Shielding

☐ Shield sides ☐ Shield top/bottom

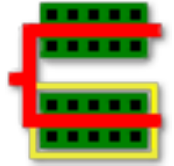
Side Width 0.62 Side Spacing 0.18

Top/Bottom Width 0.62

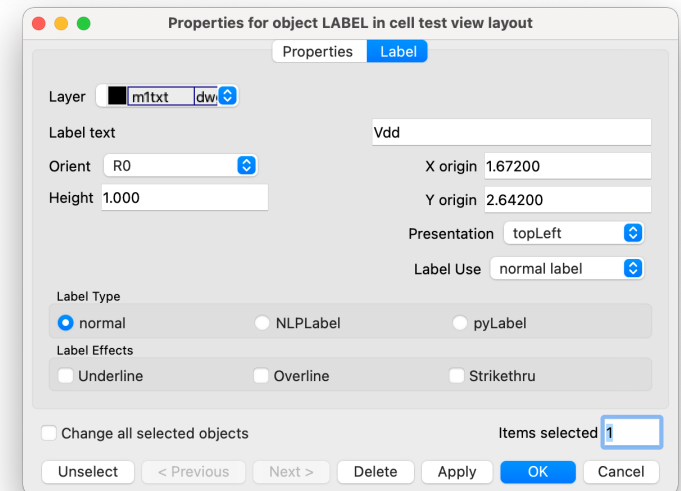
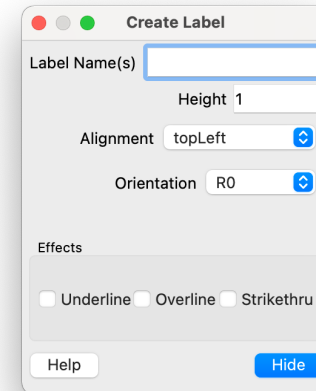
Shield Net

Help Hide

# Create Label

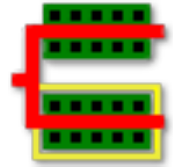


- Create Label
  - Labels are just text
  - They are not shapes
    - Can't be DRC'd
  - Origin – location
  - Orient – rotation
  - Presentation
  - Type used for schematic/symbols
    - text relative to origin

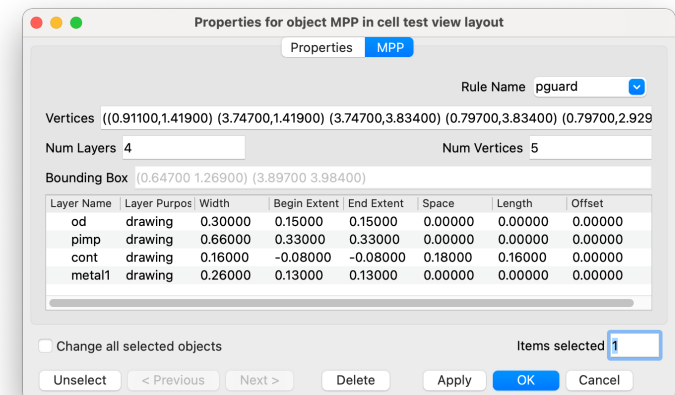
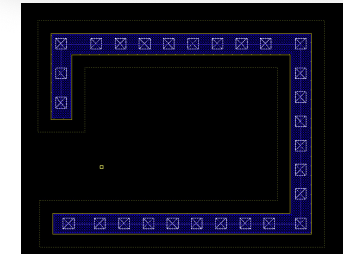
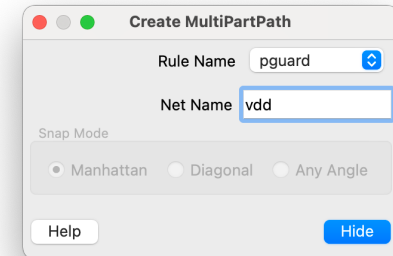


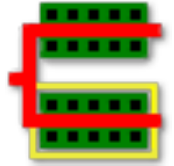


# Create MultiPartPath



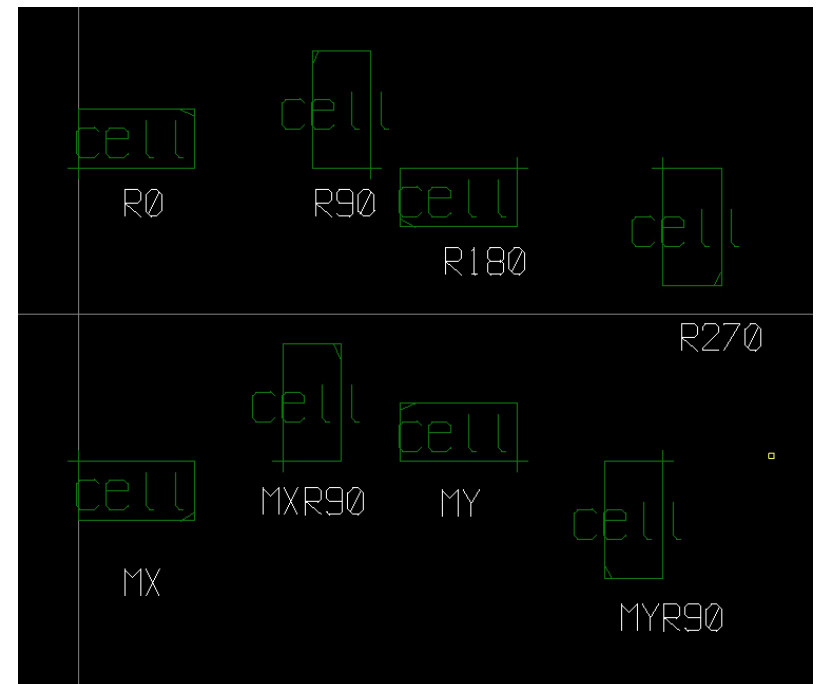
- A MPP is a complex path. It is stored as a vector of points.
- It is composed of one or more layers
  - Each layer can have a width, begin/end extent
    - Optionally space, length for contacts
    - Optionally an offset away from centre of path
- MPPs are defined in the techfile by name
- They simplify guard rings, power straps etc.
- Exported to GDS2 as shapes



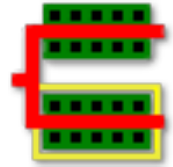


# Orientations

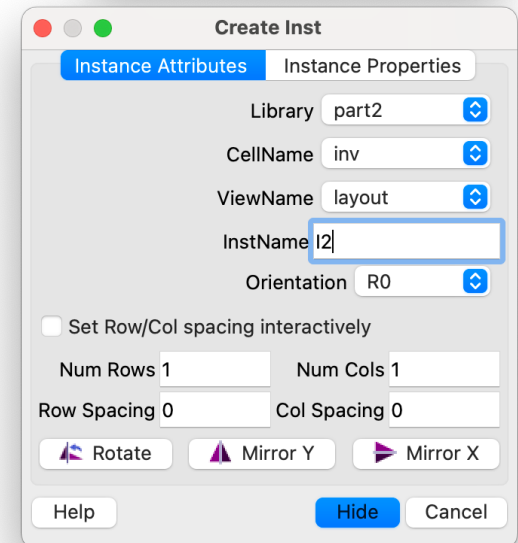
- The orientations R0, R90, R180, R270, MX, MXR90, MY, MYR90 define all possible (Manhattan) orientations



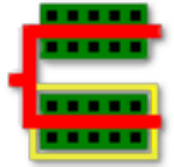
# Create Instance



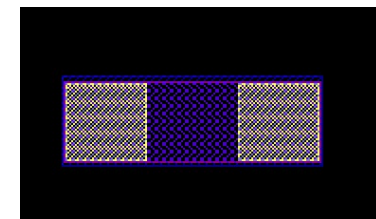
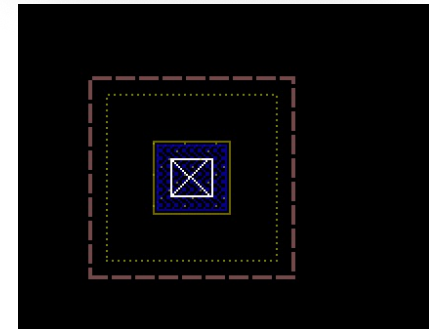
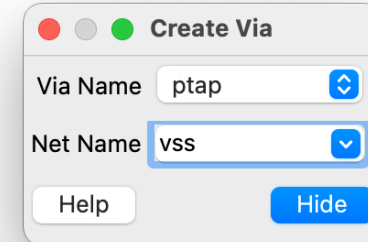
- Create Instance shows an outline of the instance, and its cell name.
- Use the option form to define instance's library/cell/view names.
- Arrays can be generated by setting number of rows/cols and spacing
- Instances current can have orientations R0, R90, R180, R270, MX, MXR90, MY, MYR90



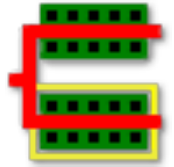
# Create Via



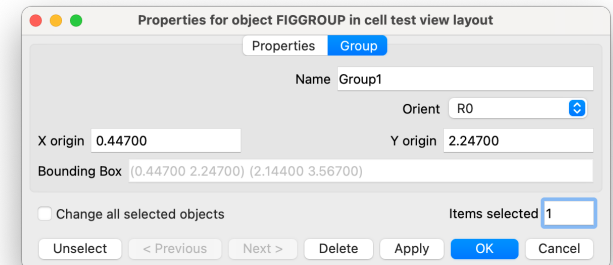
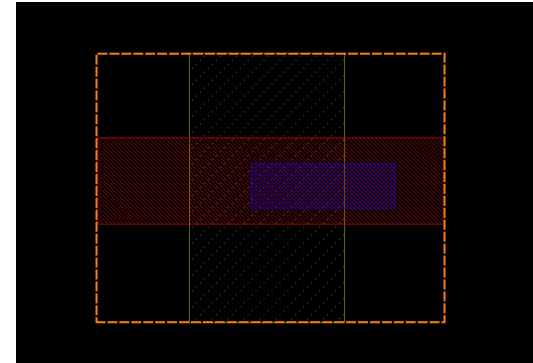
- A Via is a layer pattern defined in the techfile
- It can have multiple layers, with rectangular shapes defined relative to the origin.
- Used for connection between layers
- Used for well taps

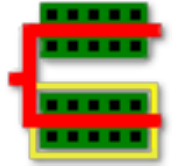


# Groups



- A group is a collection of shapes that can be manipulated as a whole
  - Kind of like an instance, but without hierarchy
  - Groups can have objects added/removed
  - Groups have an origin and an orientation
  - They can be moved and copied
  - Groups can be ungrouped



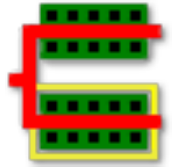


## 2. Selection

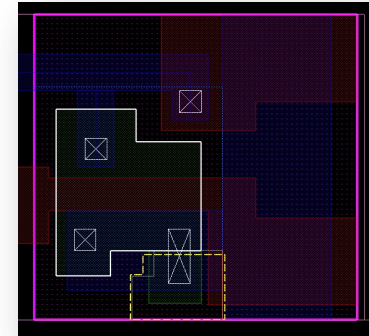
- Many commands work on the Selected Set.
  - Selected Set is a list of objects that have been selected.
  - Selected objects have their outline drawn in the ‘select’ layer (defaults to white).
  - Selection can be made by:
    - LMB (Left mouse button) click on an object
    - Shift+LMB adds to the selected set
    - Ctrl+LMB removes from the selected set
    - LMB drag will select objects wholly contained in the drag rectangle
    - LMB drag works with modifier keys Shift and Ctrl
  - Bindkey Ctrl–A selects all objects, Ctrl–D deselects all selected objects.



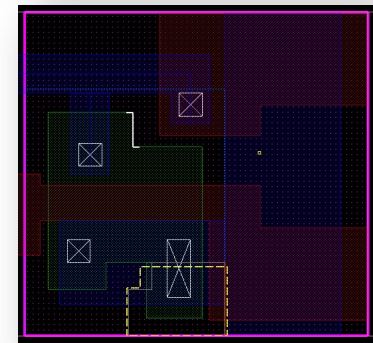
# Selection



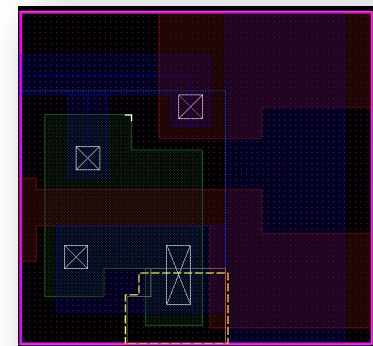
- Shapes can be selected according to the current selection mode
  - FULL: selects the whole shape
  - PARTIAL: selects either an edge of a shape or a vertex of a shape. Vertices will be selected if the cursor is within 10% of the nearest edge length to the vertex.
- Toggle between full/partial mode with the F4 bindkey
- Other objects can also be selected
  - Instances/arrays
  - Text labels



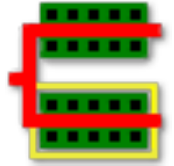
Full



Partial  
(Edge)

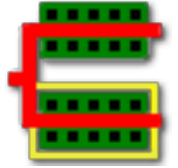


Partial  
(Vertex)



# Selection

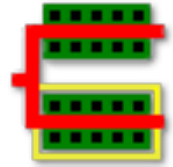
- Selection can be controlled by:
  - Making layers unselectable/invisible (can't select invisible layers)
  - Making the Instance layer unselectable prevents selecting instances or arrays
  - When there are multiple object under the cursor, if the cursor does not move and the left mouse button click repeated, objects with edges near the cursor are cycled through.
- Number of objects selected is shown in the status bar
- Query selected objects using the Q bindkey (Edit->Query)



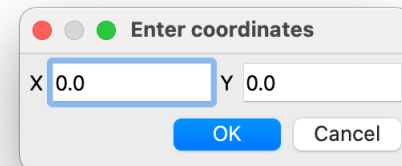
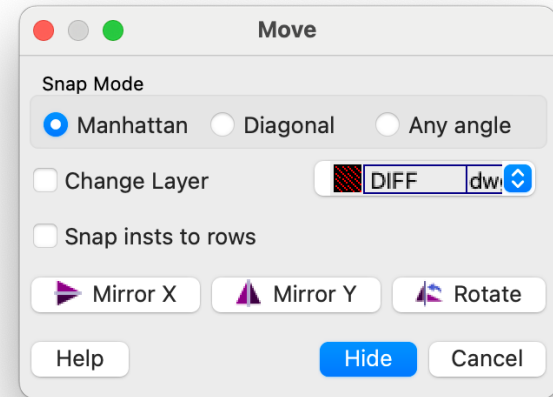
# 3. Editing

- Move, Copy, Stretch
- Yank, Paste
- Align
- Edit In Place
- Hierarchy
- Boolean operations
- Undo/Redo

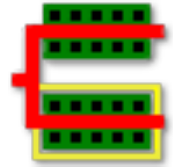
# Move



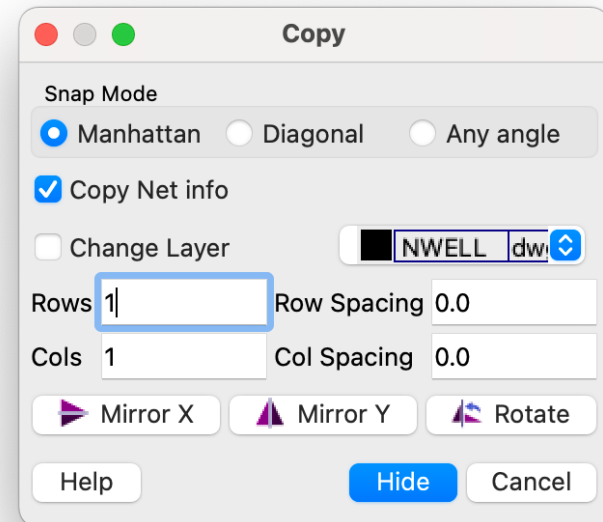
- Move
  - Select object(s) and move. First point is the reference point, second point defines the delta for the move.
  - To move a fixed distance, Use Edit->Move, then F5. Enter X/Y coords as 0,0. Then F5 again and enter delta X,Y required.
  - Move can rotate/mirror shapes
    - Use r, x, y bindkeys during move
  - Move can change the layer of the shape



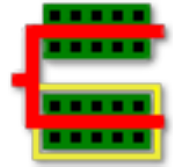
# Copy



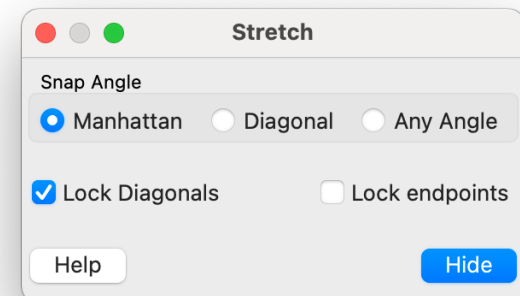
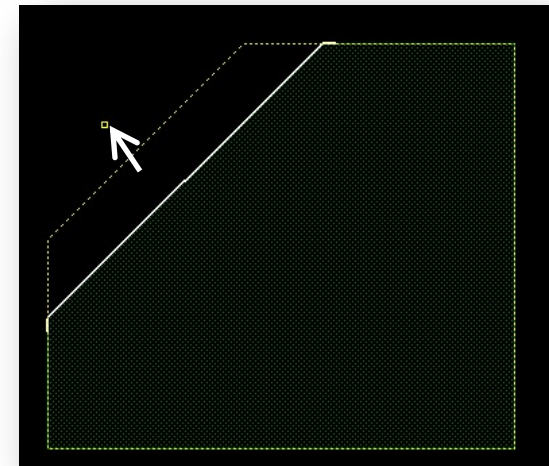
- Copy
  - Similar to move: select the object(s) to copy, then enter the reference coordinate and the destination coordinate.
  - Copy can copy the object(s) by array (not an instance array)
  - Copy can rotate/mirror during copy
    - Use r/x/y bindkeys or option form buttons
  - Copy can change layer of shape(s). All shapes will be copied to the target layer.



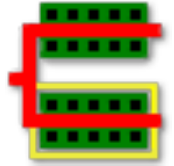
# Stretch



- Stretch
  - Select edge in partial select mode (F4)
  - Stretch will show as yellow dotted line
  - Stretch options form allows ‘locking’ diagonals. Otherwise stretching a manhattan edge of the object may change the adjacent diagonal to any angle.
  - Stretching paths can lock endpoint(s).
  - Stretch works on selected vertices as well as edges.
  - Stretch will *move* shapes that have been selected in full select mode.
    - Includes any selected instances



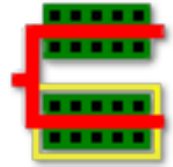




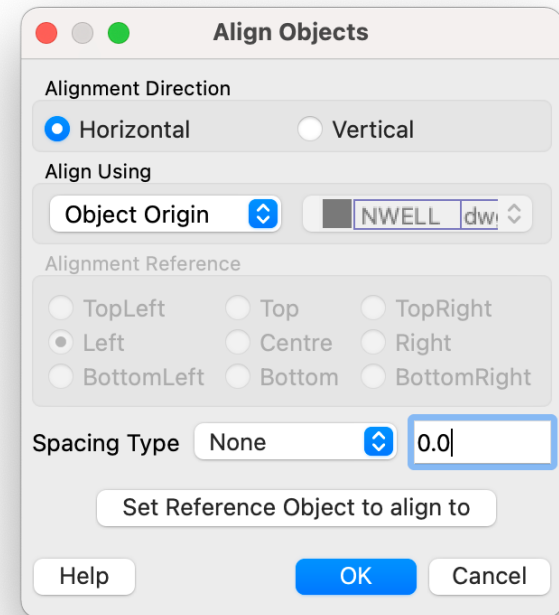
# Yank and Paste

- Copy command works only within a cellView.
- Yank (ctrl-C) and Paste (ctrl-V) will copy objects from one cellView to another (also works in same cellView)
- But more limited
  - Can't change layer, rotate, make array etc.

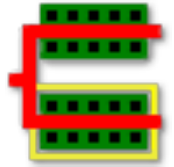
# Align



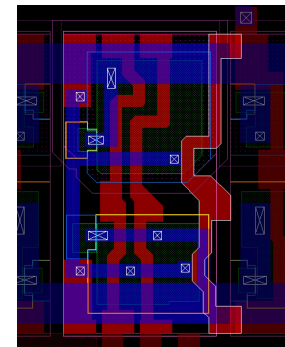
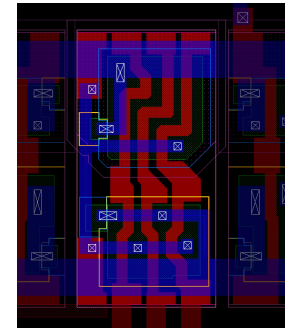
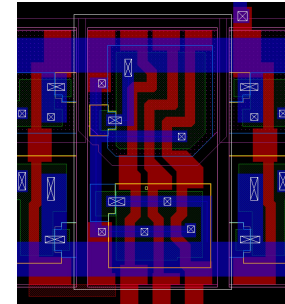
- Align
  - Powerful command to align objects
  - Horizontally or vertically
  - Align by
    - Object origin
    - Object bounding box
    - Layer bounding box
  - Optionally space objects
    - By techfile spacing rule
    - By user entered spacing
    - By user entered pitch
- To align objects,
  - Display the dialog
  - Use 'Set Reference Object'
  - Click on objects to align to reference
- Does NOT use selected set!



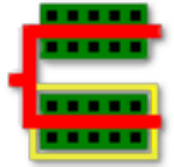
# Edit In Place



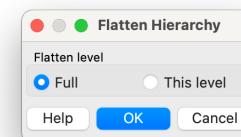
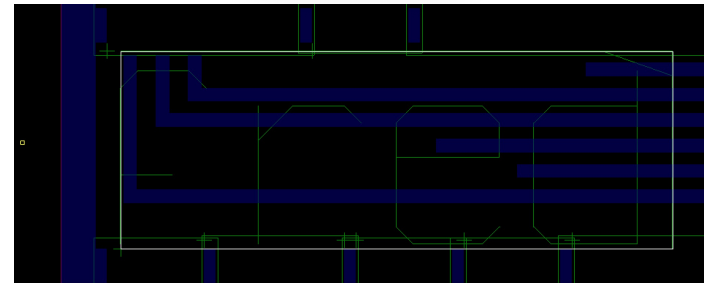
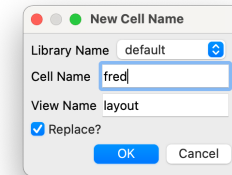
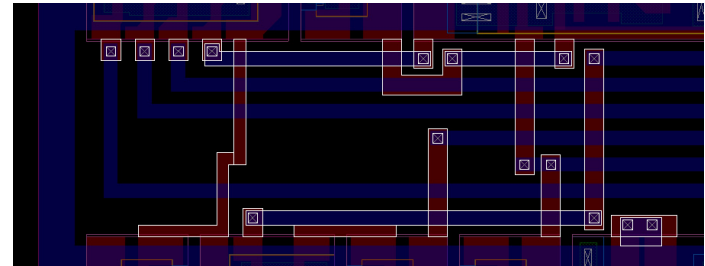
- A way of editing a cellView's contents while viewing its surroundings.
- Select an instance, Edit->Edit In Place ('[' bindkey)
  - EIP cellView is shown against dimmed background
  - **Changes made in the edit cell are reflected in all other instances of that cell!**
- You can edit in place multiple levels deep
- Return up a level at a time



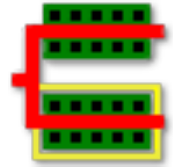
# Hierarchy



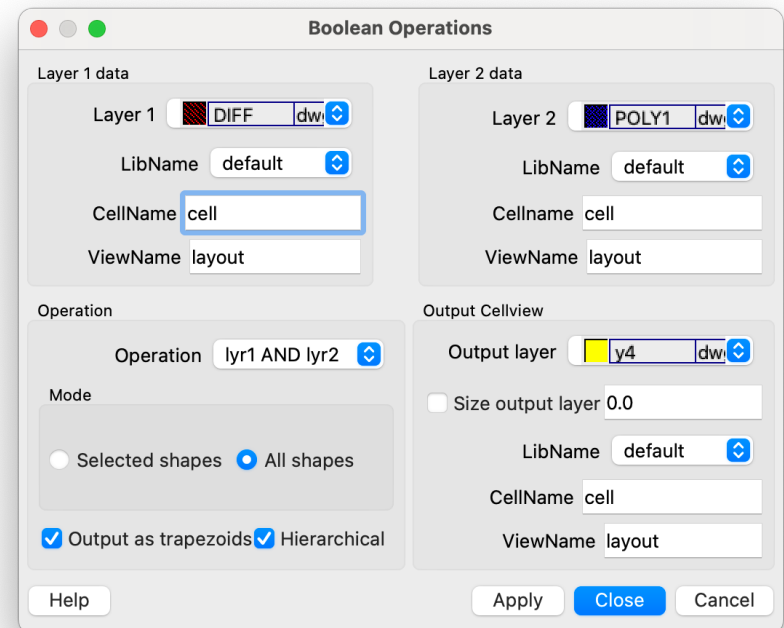
- Edit->Hierarchy menu
  - Ascend
  - Descend
  - Create CellView...
    - Selected objects are (optionally) replaced by a cellView containing those objects
  - Flatten...
    - Selected cellView is flattened
    - 1 level or all levels



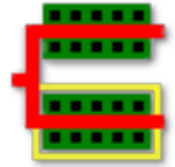
# 4. Boolean Operations



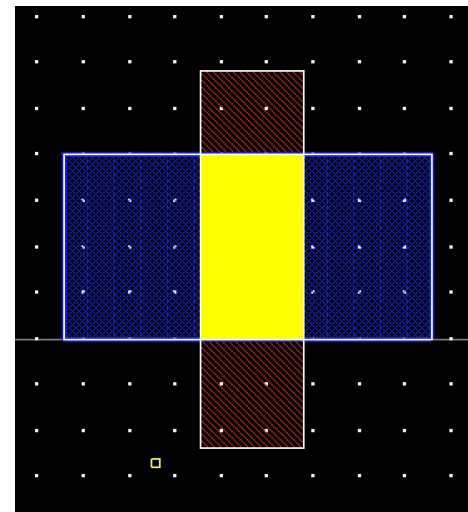
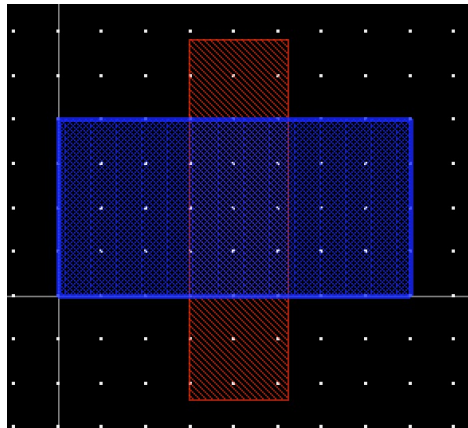
- Boolean Operations
  - One or 2 layers as input
  - Selected shapes or all shapes
  - Same or different cellViews
  - Same or different output layer and cellView
  - AND, OR, NOT, XOR, SIZE etc
  - Flat or hierarchical input
  - Decompose output shapes to trapezoids
- For large numbers of shapes, 'Tiled Booleans' command allows multithreading



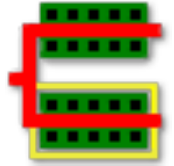
# Boolean Operations



- Example
  - Select Poly and Diff layers
  - Select the shapes
  - Use lyr1 AND lyr2
  - Output to y4 (yellow)







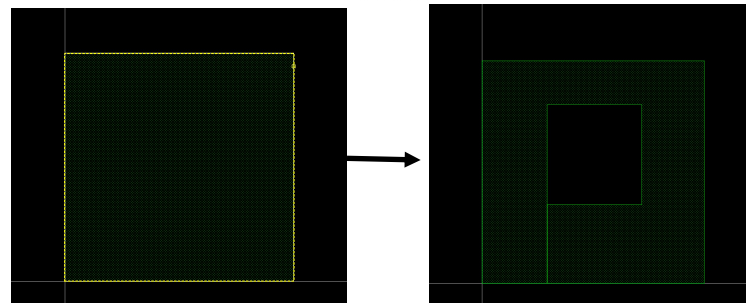
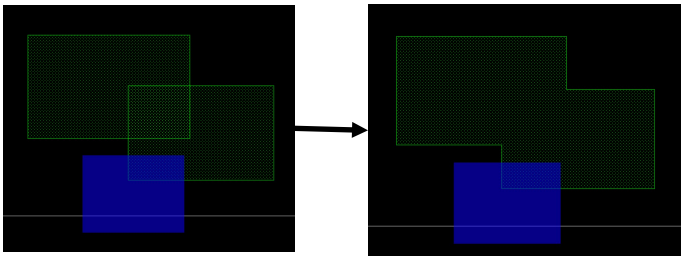
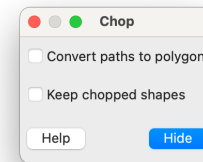
# Merge and Chop

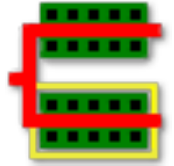
- Merge

- Takes selected shapes
- Merges all shapes on same layer

- Chop

- Takes selected shape
- Cut a rectangle in shape
- Paths can be maintained
- Chop shapes can be kept

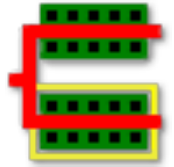




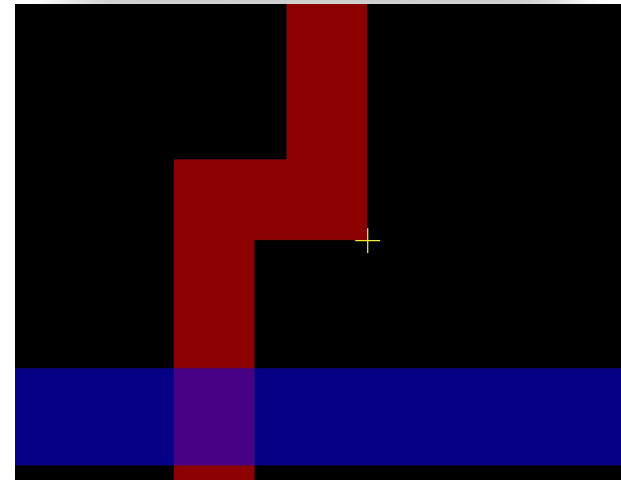
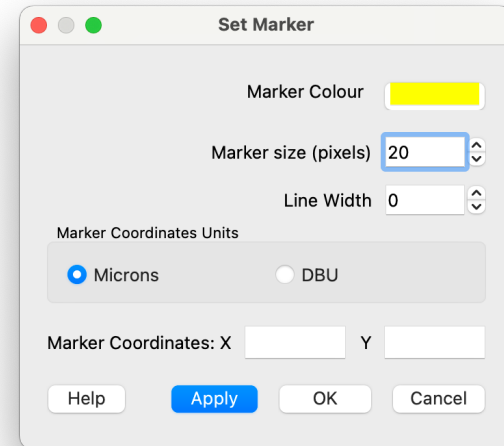
# Undo/Redo

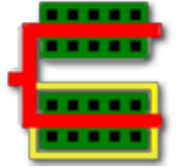
- Can undo and redo many, but not all, operations
  - E.g. Booleans can't be undone
  - Undo and redo stack – currently 16 levels

# Markers



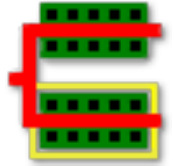
- Tools->Add Marker...
  - Set marker color/size/linewidth
  - Set units in um or dbu
- Tools->Clear Markers
  - Clears all





# 5. PCells

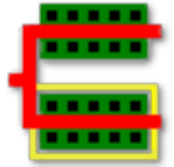
- What are Pcells
- How do you create Pcells
- How are Pcells Used



# What are PCells

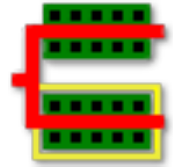
- PCells are Parameterised Cells
  - Instances of a PCell can be different depending on the properties of the instance. For example a MOS device can have W, L parameterisable.
- Benefits of PCells
  - Designed once – can be used in many variants
  - Reduce DRC errors
  - Faster layout times
- Glade PCells are NOT the same as Cadence Pcells or Synopsys PyCells!
  - Other vendor's PCells use proprietary languages (e.g. Skill) and/or platform-dependent plugins (PyCells)

# How do you create PCells



- Glade PCells written in Python
  - Code can be debugged using print statements or using a Python debugger e.g. pdb.
  - PCell files must be kept in a directory in user's PYTHONPATH
  - Bytecode compiled PCells can be used (.pyc files) to distribute unreadable PCells.
- PCell code creates a cell called a SuperMaster. This cell is used to create instances of PCells. When an instance of the supermaster is created, a SubMaster cell is also created using the unique properties of the instance.
- SubMaster cells are not visible in the library browser
  - they are managed by the PCell subsystem.

# How do you create PCells

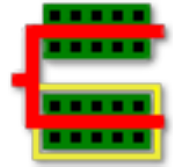


```
# Import the db wrappers
from ui import *
```

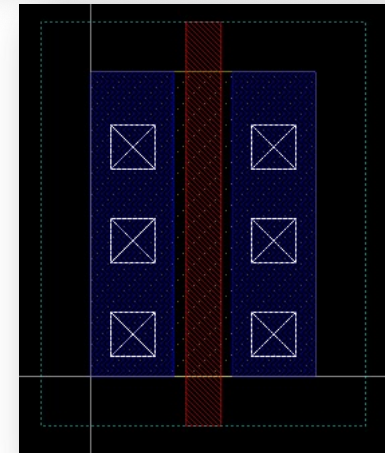
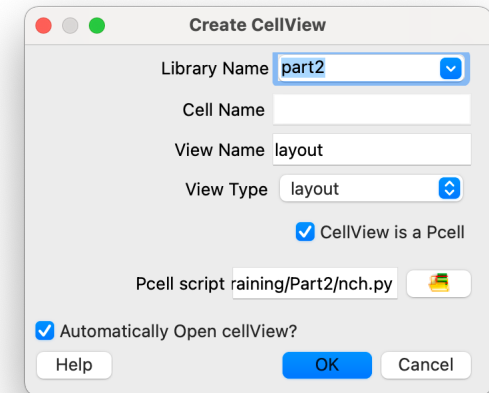
```
# The entry point. The function name *must* match the filename.
def nmos13_multi(cv, w=1.1, l=0.13, m=1) :
    lib = cv.lib()
    dbu = lib.dbuPerUU()
    width = int(w * dbu)
    length = int(l * dbu)
    fingers = int(m)
```

- An example: nmos13\_multi.py (only the first few lines shown)
  - We define a function called nmos13\_multi.
  - It has 4 arguments
    - 1<sup>st</sup> is always the cellView that the PCell instance is created in.
    - Remainder are the PCell parameters. They \*must\* have default values specified, so we can build the PCell if the properties have not yet been specified on the PCell instance

# How are Pcells Used

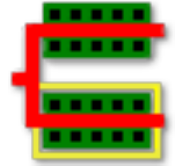


- Create a Pcell supermaster manually using the New Cell command.
- This creates the SuperMaster cell `nmos13_multi`
  - Do not edit this cell!
  - It is there for 2 reasons
    - To help you debug writing pcells
    - To allow the Create Instance command to reference it. An instance of its submaster is then created and used.





# How are Pcells Used



- To create a PCell instance in your layout:
  - Use the Create Instance command
  - Edit the Instance Properties tab on the options form to set the PCell parameters.
- The Properties tab shows the default Pcell parameters; you can modify these to the values you want.

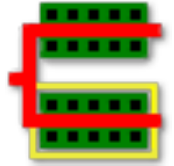
The 'Create Inst' dialog box, 'Instance Attributes' tab, shows the following fields and controls:

- Library: part2
- CellName: pch
- ViewName: layout
- InstName: I2
- Orientation: R0
- ☐ Set Row/Col spacing interactively
- Num Rows: 1
- Num Cols: 1
- Row Spacing: 0
- Col Spacing: 0
- Buttons: Rotate, Mirror Y, Mirror X
- Buttons: Help, Hide, Cancel

The 'Create Inst' dialog box, 'Instance Properties' tab, shows a table of parameters:

Name	Type	Value	Callback	Display
type	string	mos		true
w	float	1.1e-06		true
l	float	1.3e-07		true
polyCnt	boolean	false		true
leftCnt	boolean	true		true
rightCnt	boolean	true		true

Buttons: Help, Hide, Cancel



# 6. Labs

- Create an inverter cellView using Pcells
- Glade\_training/Part2