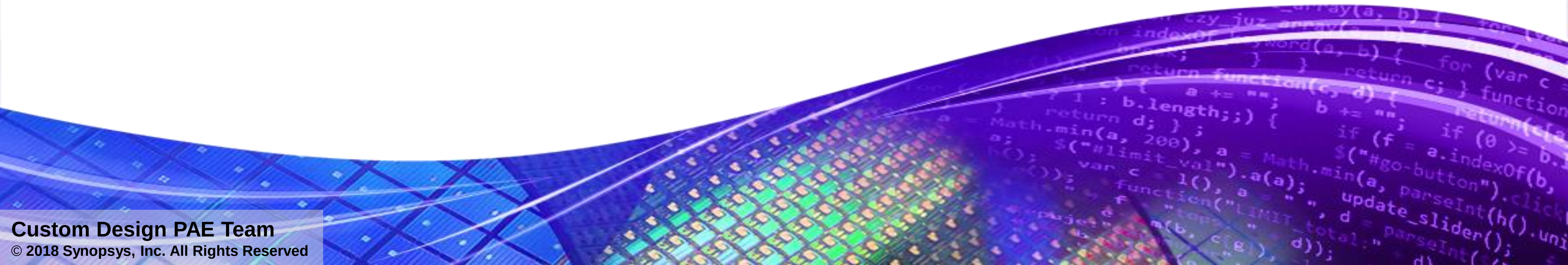
Custom Compiler

Layout Editor (LE)
Advanced Editing Functions

O-2018.09

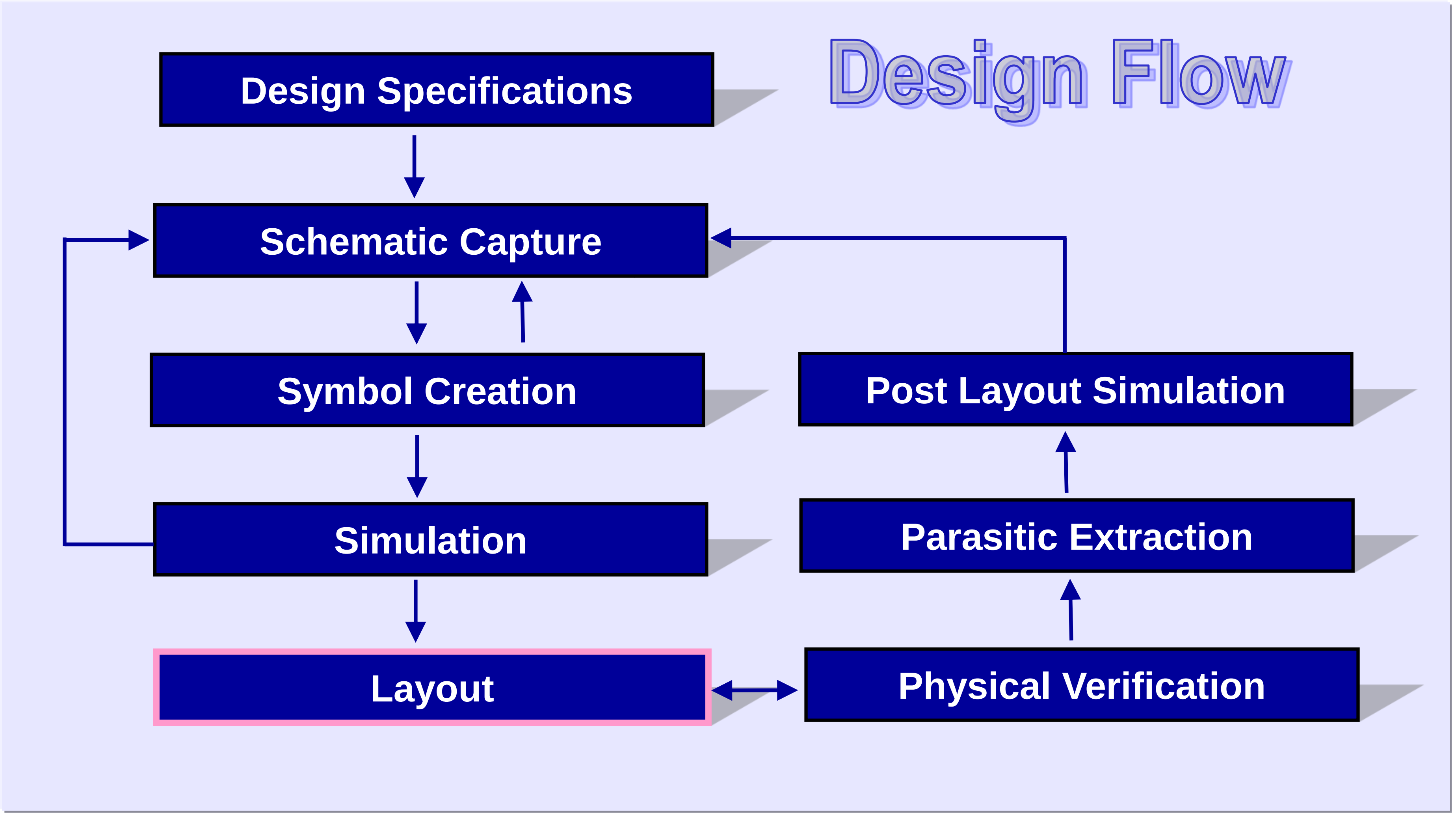


Custom Design PAE Team

Unit Objectives

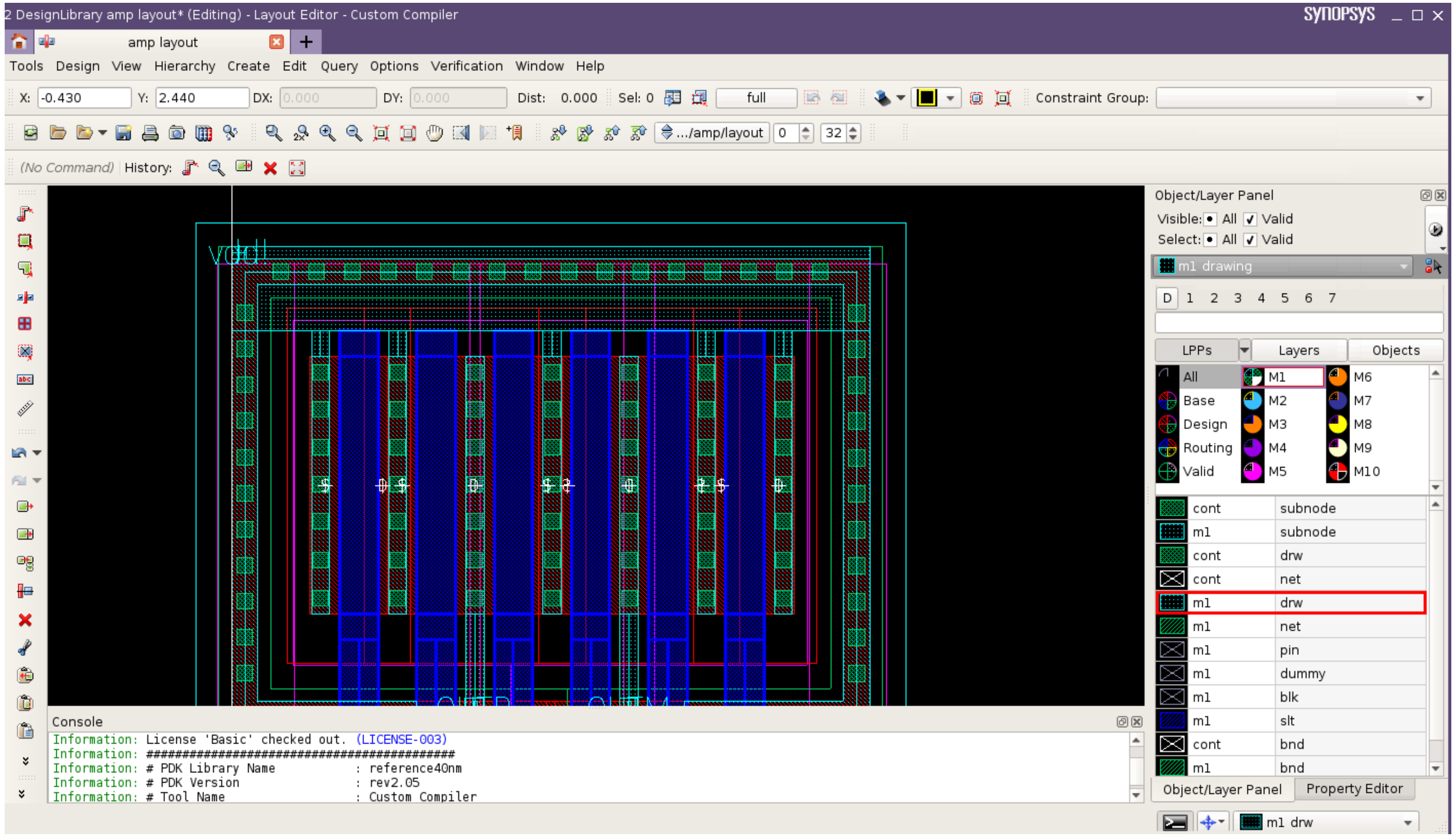
- **After completing this unit, you will be able to:**
  - Use different modes for data creation & editing functions
  - Stripe Wires
  - Use Cut Poly command

Full Custom Compiler Flow



Layout Editor Console

- **User can input commands from Console assistant invoked right in Layout Editor window**



2 DesignLibrary amp layout* (Editing) - Layout Editor - Custom Compiler

X: -0.430 Y: 2.440 DX: 0.000 DY: 0.000 Dist: 0.000 Sel: 0 full Constraint Group:

.../amp/layout 0 32

(No Command) History:

Visible: ● All ☑ Valid
Select: ● All ☑ Valid

m1 drawing

cont	subnode
m1	subnode
cont	drw
cont	net
m1	drw
m1	net
m1	pin
m1	dummy
m1	blk
m1	slt
cont	bnd
m1	bnd

m1 drw

Console

```
Information: License 'Basic' checked out. (LICENSE-003)
Information: ###########################################
Information: # PDK Library Name            : reference40nm
Information: # PDK Version                 : rev2.05
Information: # Tool Name                   : Custom Compiler
```

Selection Use Models

- **Pre Selection**
  - Select the objects and activate the command
  - Command is one-shot and selected set remains
- **Post Selection**
  - Activate the command and select the objects
  - Command is modal and selected set becomes de-selected
- **Post Selection with Infix**
  - If infix enabled and active object exists, first point defines the reference point for the command

Infix Mode Use Model

- **What is Infix mode ?**
  - Applicable when a command is initiated with bindkey
  - Edit commands operate on active object (no selection)
  - Create commands start from cursor location
- **How to set**
  - Options > General dialog
  - Preference deInFixMode: true | false (default)
- **Non-Infix with/without bindkey**
  - Activate the command
  - LMB to define initial point

Direct Manipulation: Stretch/Copy

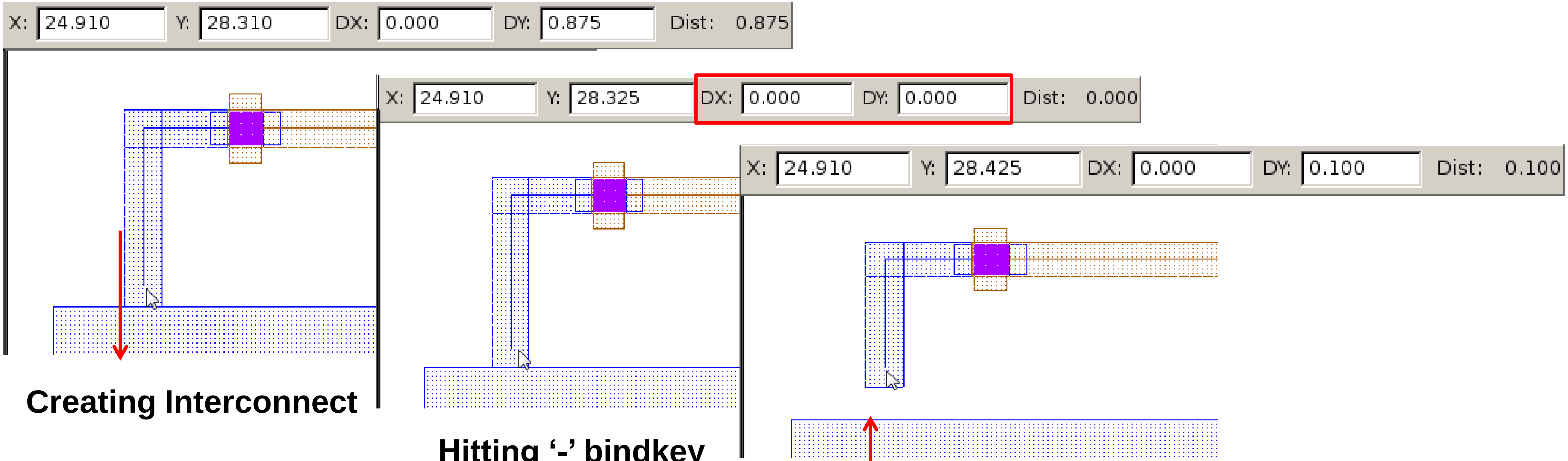
- **Direct Manipulation**
  - Move/copy/stretch operations on objects by cursor drag
  - Controlled by two preferences

leDirectManipulation	true false (default)	Enable or disable
leDirectManipulationSelectedOnly	true (default) false	Execute when active object is selected

BindKey	Description	Action
Button1 + Drag	Start stretch on selected or active object(s)	leStartStretchDrag
Ctrl + Button1 + Drag	Start copy on selected or active object(s)	leStartCopyDrag

Reset DX/DY Relative Distances

- **Toolbar's Dx/Dy can be reset to 0,0 while editing**
  - Ex.: during Create Interconnect, Move, Stretch
  - Binding "-" → Redefines the anchor point to be the actual point
  - Binding "0" → Resets the anchor point back to the original location



X: 24.910 Y: 28.310 DX: 0.000 DY: 0.875 Dist: 0.875

Creating Interconnect

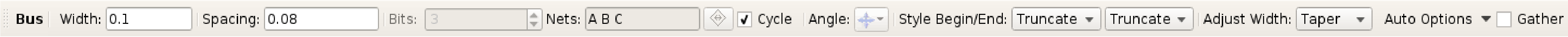
X: 24.910 Y: 28.325 DX: 0.000 DY: 0.000 Dist: 0.000

Hitting '-' bindkey

X: 24.910 Y: 28.425 DX: 0.000 DY: 0.100 Dist: 0.100

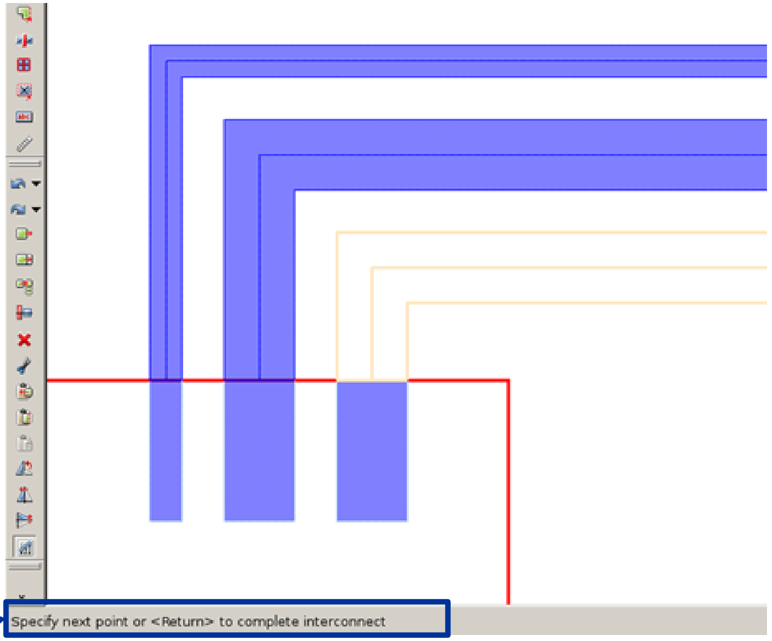
Pulling to desired spacing

Bus Creation: Bus command options



Bus Width: 0.1 Spacing: 0.08 Bits: 3 Nets: A B C ☑ Cycle Angle: Style Begin/End: Truncate Truncate Adjust Width: Taper Auto Options □ Gather

- **Create Bus command supports per-pin width and spacing**
- **Use Model:**
  - Start Create Bus command
    - Bits: # bits in bus
    - Select Angle: Orthogonal, Diagonal, X-First, Y-First
    - Adjust width: Taper
    - Auto Options: Terminate = Off
  - Select the first pin
  - Shift-dragButton1 select all other pins of the bus
  - Route to the destination pin
  - Shift-dragButton1 select all destination pins
- **Best use with pathSegs for different source and destination pin widths**
  - Note: Instructions are located in the Message bar

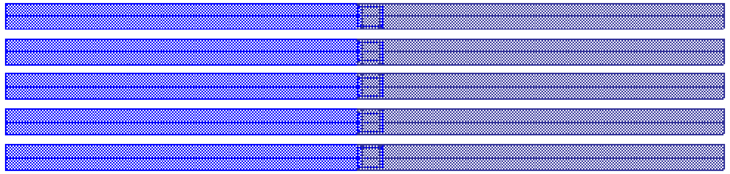


Specify next point or <Return> to complete interconnect

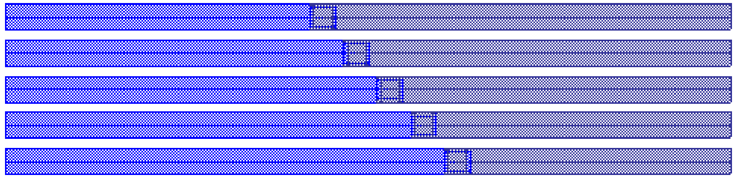
- **Gather option**
  - Defines if always use minSpacing of two adjacent bits during bus creation.
  - If this option is True, spacing option will be disabled to forbid user from changing spacing value.

Bus Creation: Via Templates

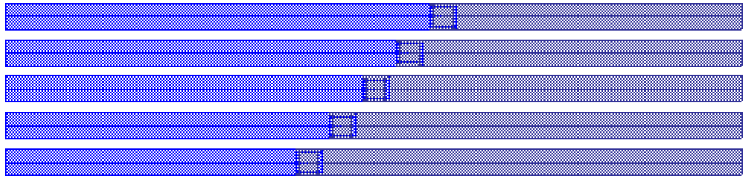
Straight



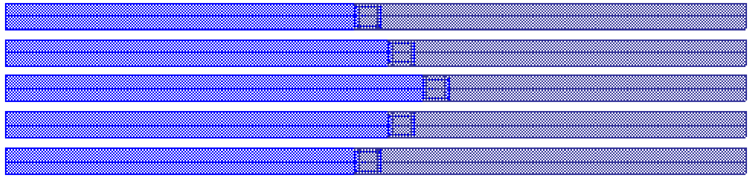
Backslash



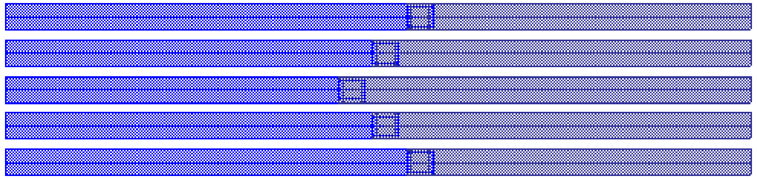
Slash

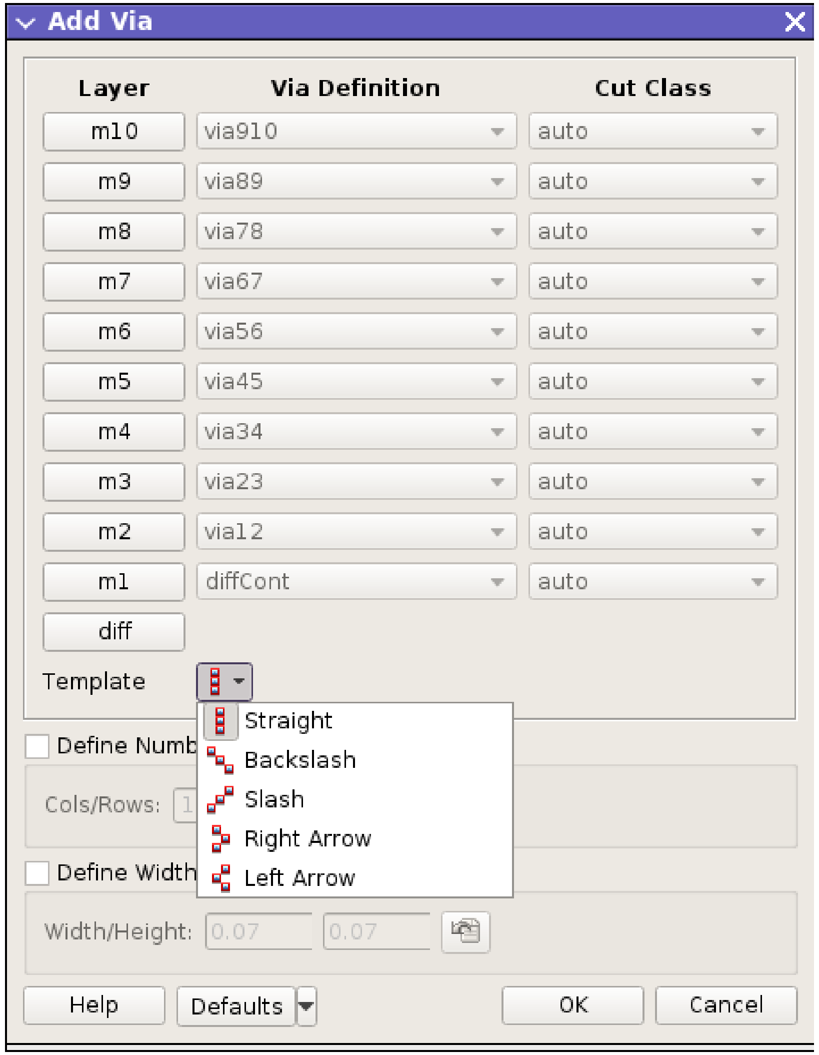


Right Arrow



Left Arrow





Add Via

Layer	Via Definition	Cut Class
m10	via910	auto
m9	via89	auto
m8	via78	auto
m7	via67	auto
m6	via56	auto
m5	via45	auto
m4	via34	auto
m3	via23	auto
m2	via12	auto
m1	diffCont	auto
diff		

Template

□ Define Numb

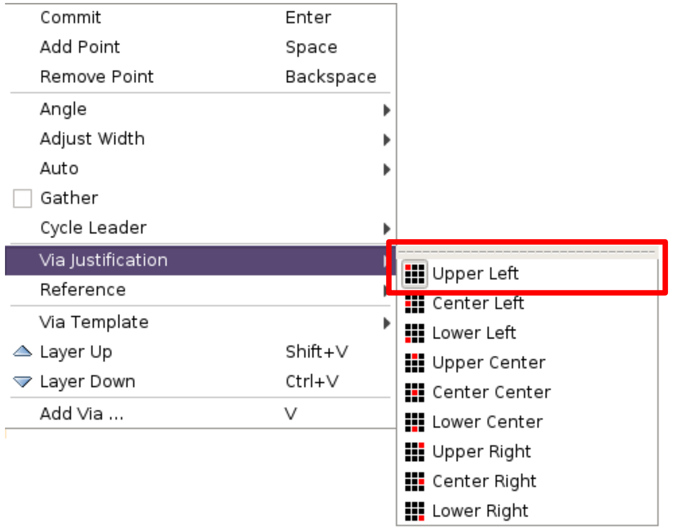
Cols/Rows: 1

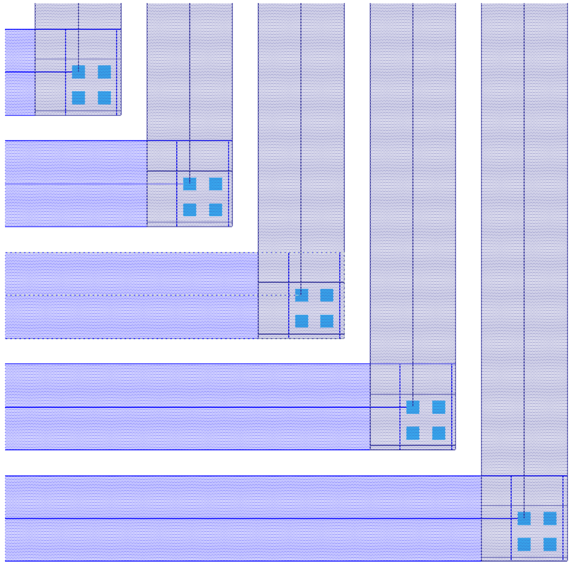
□ Define Width

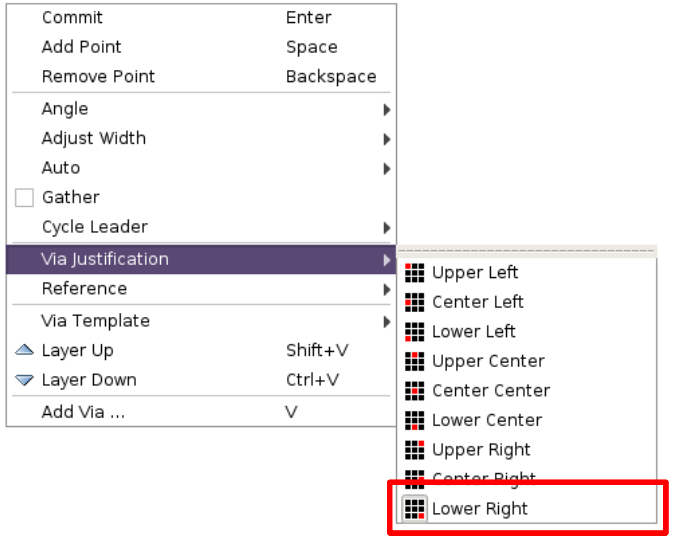
Width/Height: 0.07 0.07

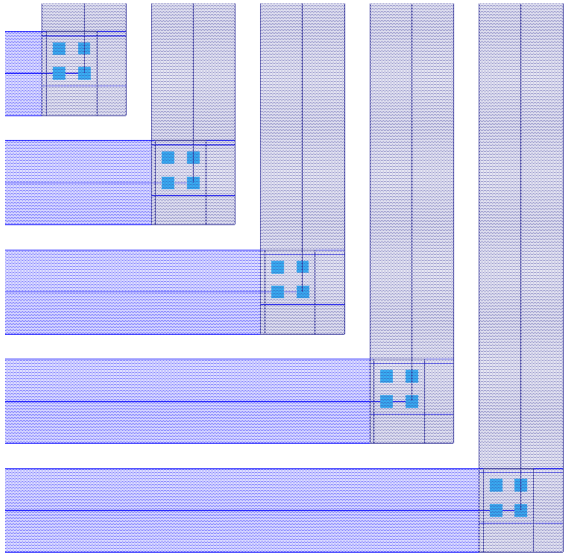
Bus Creation: Via Justification

- **Allows to specify justification for Vias when change layer during bus creation**



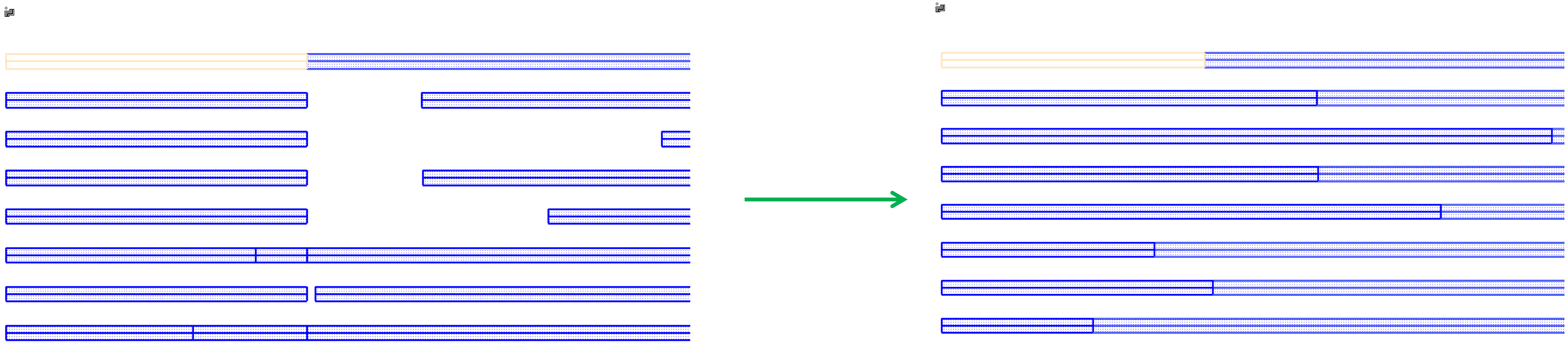






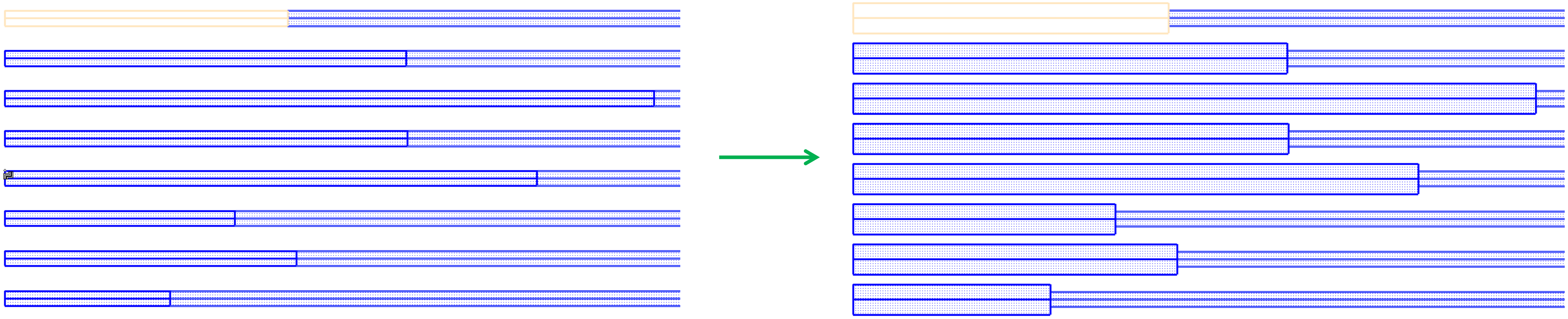
Bus Creation: Non Aligned Shapes Support

- **Works only with Adjust Width = Taper**
- **Use Model**
  - Click on first bit to start bus
  - Use Shift+Button1+drag to draw a selection window around pins
    - Use Shift-Button1 click to select additional pins
    - Use Ctrl-Button1 click to deselect pins
  - Proceed with bus creation



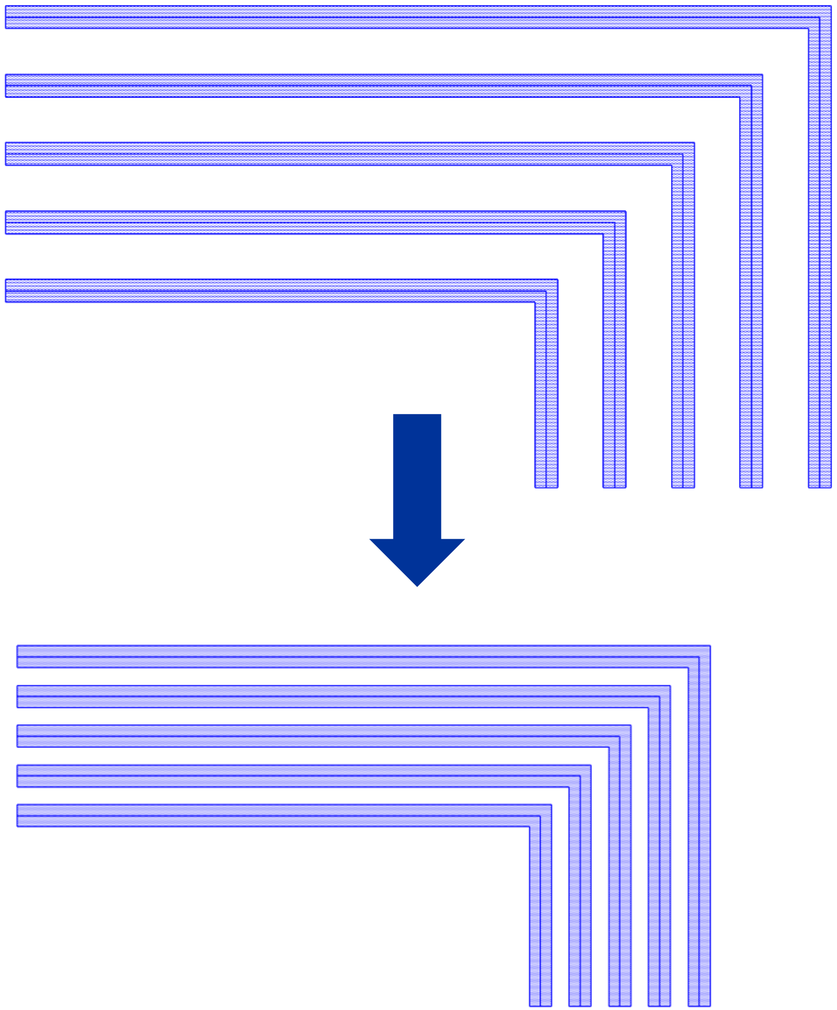
Bus Creation: NDR Support

- **Only using ‘Adjust Width’ = Taper**
  - Create Bus honors NDR after first turn
  - Other modes honor NDR for first segment too
- **In all ‘Adjust Width’ modes**
  - User can cycle through widths using Ctrl-4 or Shift-4

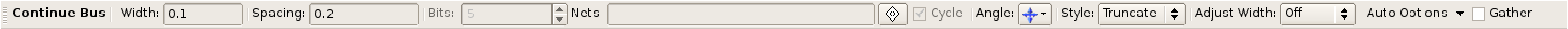


Compact Bus

- **Allows to compact bus bits to reduce spacing between bus interconnects to frees space**
  - Follow DRC rules in technology
  - Avoid 0x shapes as obstacles
    - Compacting will be ignored in case of existing 0x shape between bus bits
  - Use model:
    - Select bus interconnects
    - Call Compact Bus command
    - Use Apply to compact bus interconnects

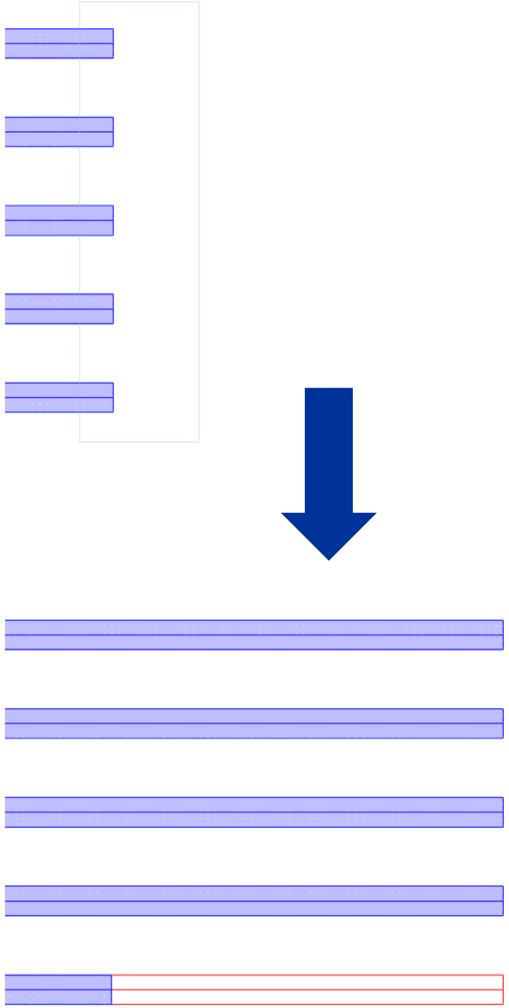


Continue Bus

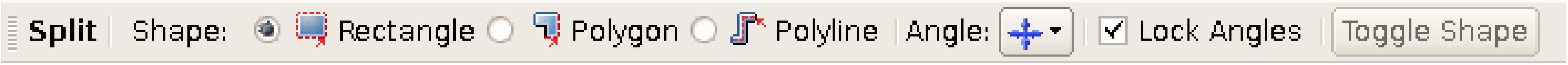


Continue Bus Width: 0.1 Spacing: 0.2 Bits: 5 Nets: ☑ Cycle Angle: Style: Truncate Adjust Width: Off Auto Options ▾ □ Gather

- **Allow to continue bus wire as solid bus**
  - Works as Stretch with bus creation capabilities
  - Has same options as Create Bus
  - Use Model:
    - Invoke Continue Bus command
    - Select bus bits edges by region select
    - Click LMB to start bus creation from that point
    - Adjust option

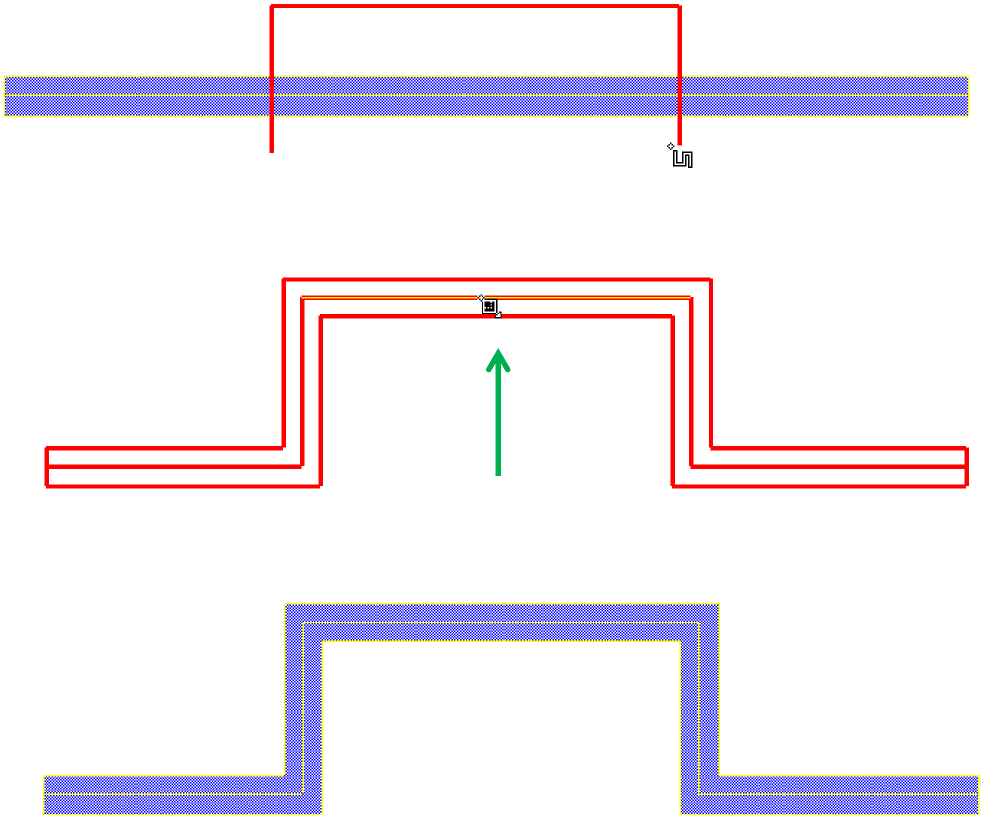


Split

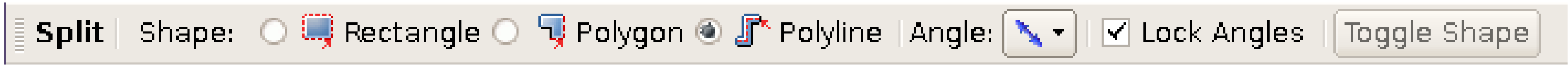


Split | Shape: ● Rectangle ○ Polygon ○ Polyline | Angle: | ☑ Lock Angles | Toggle Shape

- **Used to add segments to existing shapes or paths**
  - Menu Edit → Split → Split
  - Bindkey ‘Ctrl-S’
  - Use Model
    - Select objects to operate on
    - Draw the separation shape
    - Space key to Toggle Shape
    - Click Button1 & drag
  - Snaps to tracks

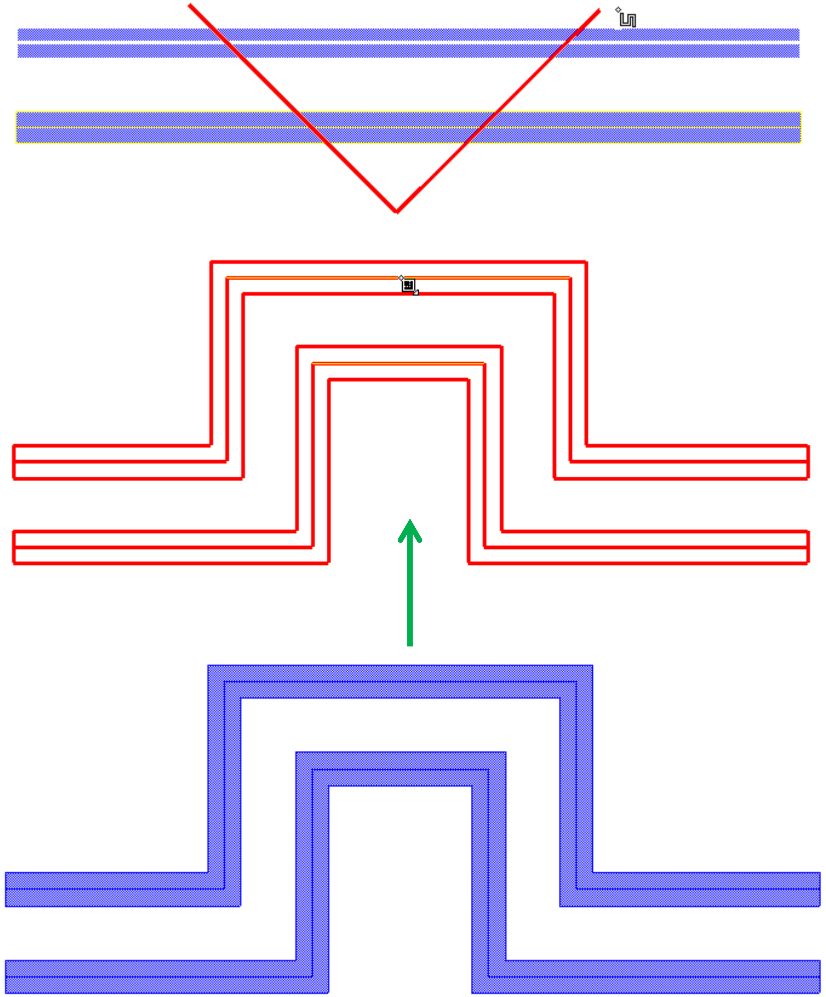


Split



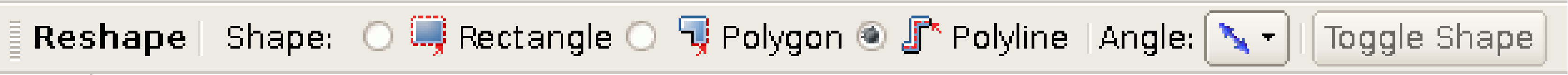
Split | Shape: ○ Rectangle ○ Polygon ● Polyline | Angle: | ☑ Lock Angles | Toggle Shape

- ■ **Used to add segments to existing shapes or paths**
  - Menu Edit → Split→ Split
  - Bindkey ‘Ctrl-S’
  - Use Model
    - Select objects to operate on
    - Draw the separation shape
    - Space key to Toggle Shape
    - Click Button1 & drag
  - Snaps to tracks



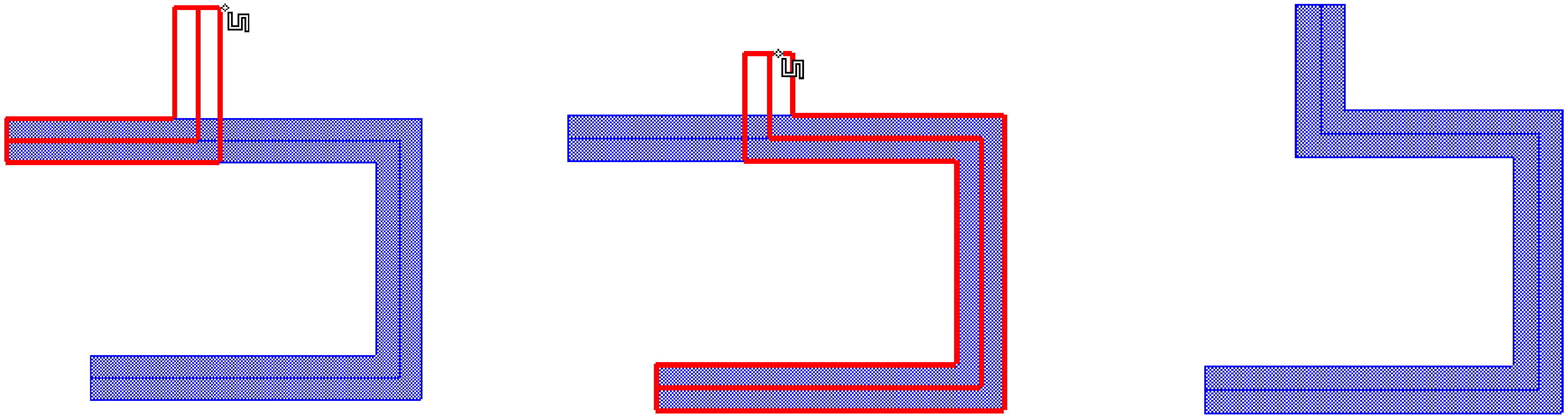
Reshape Shape

- **Edit → Reshape → Shape**
  - Bindkey ‘Shift-R’

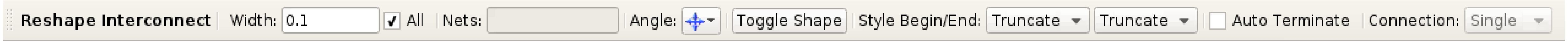


Reshape | Shape: ○ Rectangle ○ Polygon ● Polyline | Angle: | Toggle Shape

- **Allows to modify the object shape’s point list**
  - Polyline only for Paths and PathSegs
- **Use Toggle Shape to select the desired topology**
- **Snaps to tracks**

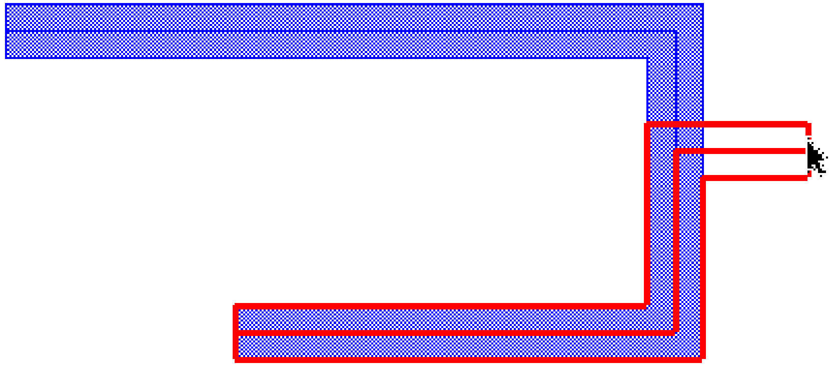


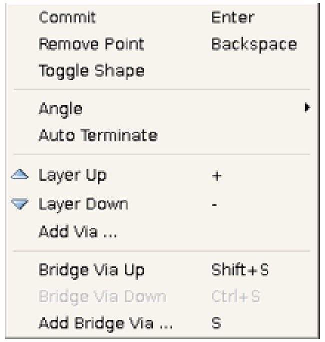
Reshape Interconnect

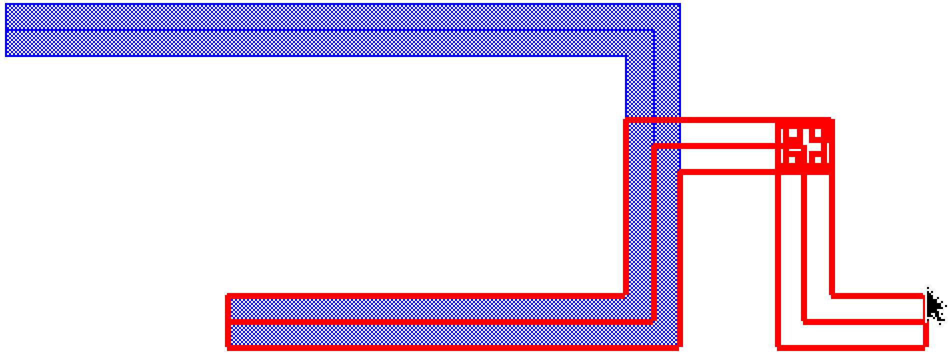


Reshape Interconnect | Width: 0.1 | ☑ All | Nets: | Angle: | Toggle Shape | Style Begin/End: Truncate ▾ Truncate ▾ | □ Auto Terminate | Connection: Single ▾

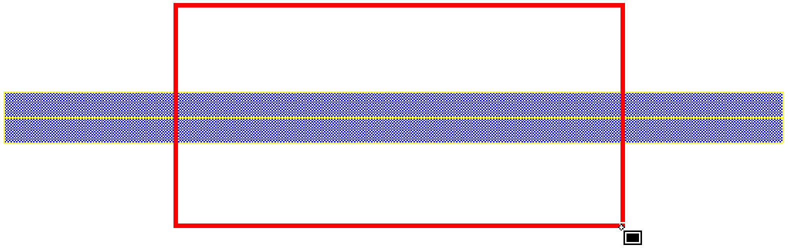
- **Edit → Reshape → Interconnect**
- **Allows to modify the object shape's point list**
  - Bindkey 'g' to set gravity on, snaps to path's center line
  - Toggle Shape with CSM or 'Shift-T'
  - Change layer same as Create Interconnect
  - Width of path can be adjusted
  - Supports multiple layers interconnect
    - Supports PathSegs only
  - Snaps to tracks
  - Supports bridge via during the interconnect reshaping
    - The Bridge Via definition in technology are required.
    - Use the CSM menu or bind keys Shift+S / Cntr+S / S to apply bridge via



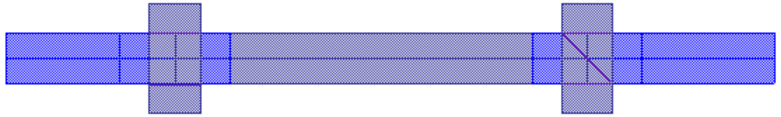




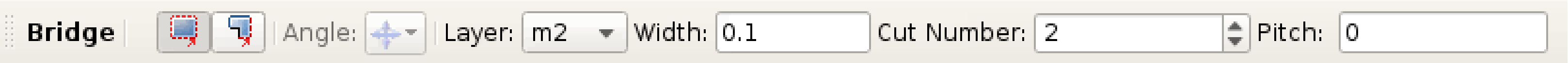
Bridge Interconnect



- **Edit → Wire → Bridge Interconnect**
  - Bindkey ‘=‘
- **Allows to change layer for an existing path**
- **Use Model**
  - Select Object
  - Draw window



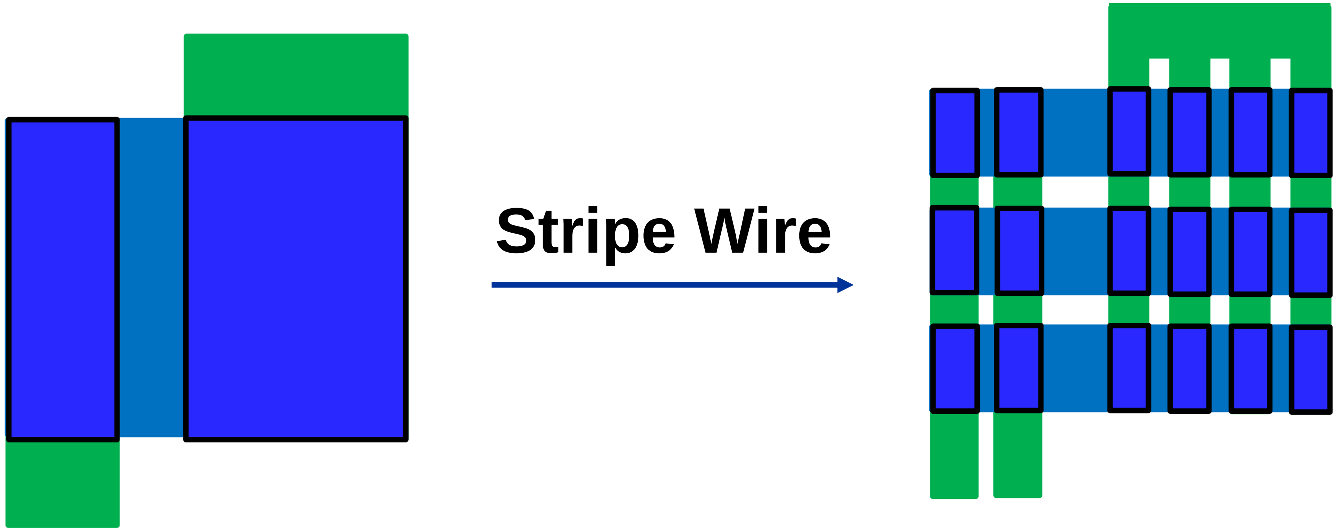
- **Tool changes layer Up or Down**
  - leBridgeInterconnectDirection
    - Above (default) | below



Bridge | Angle: | Layer: m2 | Width: 0.1 | Cut Number: 2 | Pitch: 0

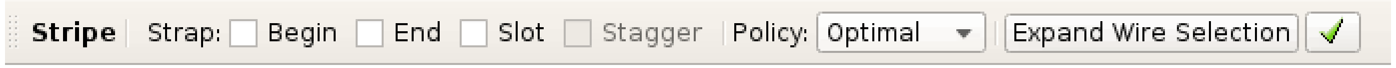
Stripe Wire Command

- **What is it?**
  - Converts fat/wide wires to striped wires
    - Design constraints
    - Physical manufacturing constraints.
  - Convert wires to a parameterized deFigure
    - Pre and post selection
  - Example
    - High current == Wide wire > maxWidth



Stripe Wire Command

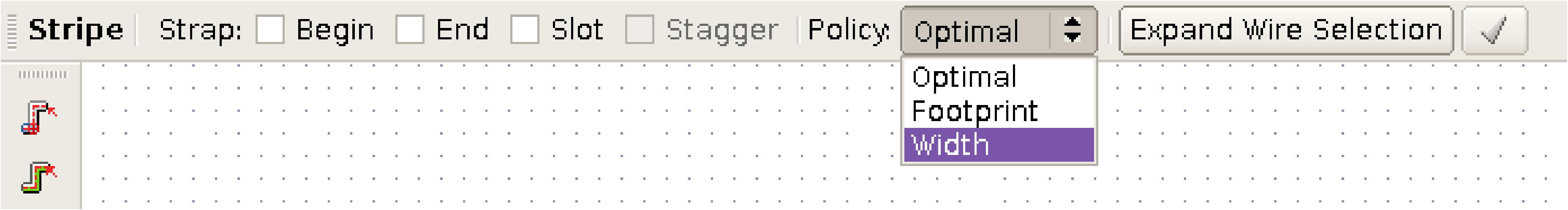
- **Use model**
  - Invoked by:
    - Edit → Wire → Stripe Wire
    - ile::stripeWire TCL command
  - Begin Strap
    - Defines whether to add a strap at the beginning of the wire preference: leSlotBeginStrap
  - End Strap
    - Defines whether to add a strap at the ending of the wire, preference: leSlotEndStrap
  - Slot and Stagger
    - Defined whether to add slots in a striped wire, preference: leSlot
    - Define if slots should be placed with an offset, preference: leSlotStagger
  - Expand Wire Selection
    - Selects the shapes by expanding the connection
- **45 degree paths/pathSegs not supported**



Stripe Strap: □ Begin □ End □ Slot □ Stagger Policy: Optimal

Stripe Wire Command

- **Stripe Wire engine works in the following way:**
  - No maxWidth constraint for the specified layer
    - Not striped
    - Converted to parameterized deFigure
  - maxWidth and minSpacing constraints for the specified layer:
- **Three striping policy:**
  - Optimal
  - Footprint
  - Width



Stripe Wire: Striping Policy

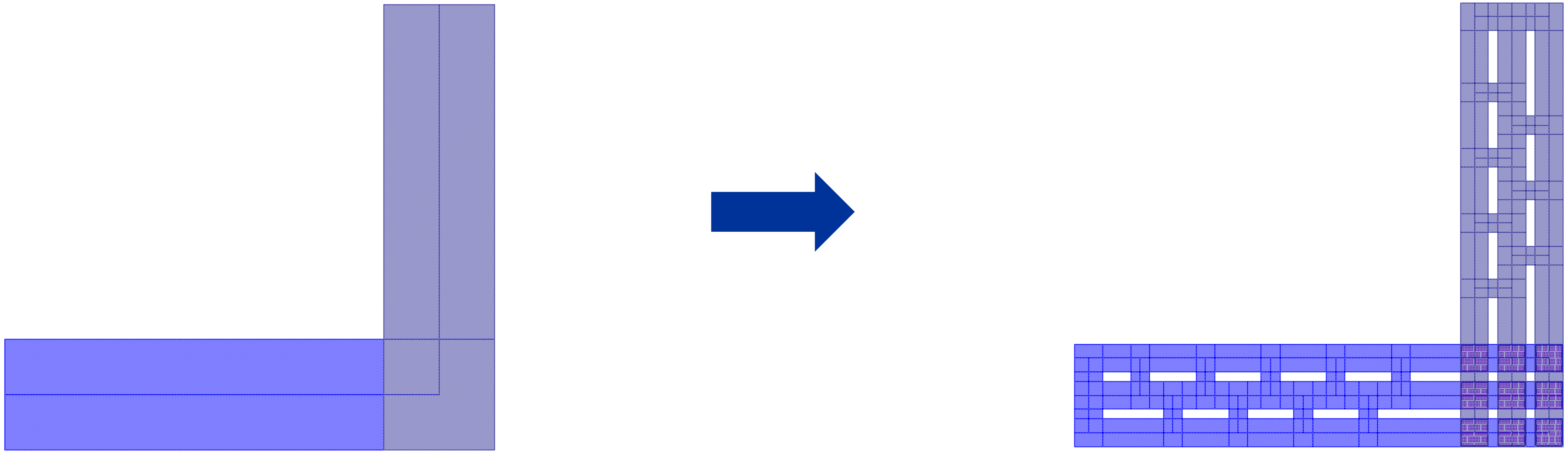
- **Three Wire Striping policy:**
  - Optimal (default)
    - Maximize the material area within the original footprint

Wire Width	Results
<= maxWidth	Not striped Converted to parameterized deFigure
> maxWidth && < 2*maxWidth + minSpacing	Two stripes
> 2*maxWidth + minSpacing	N stripes where N=(W-maxWidth)/(maxWidth+minSpacing)+1

  - Footprint
    - Keeps the same footprint with the smallest number of stripes
  - Width
    - The total width of the striped wire will be equal to the width of the original wire
- **Preference: leStripePolicy**
  - Values: optimal | footprint | width

Stripe Wire Command

- **Engine Operation Results**
  - Some examples of Stripe Wire engine operation

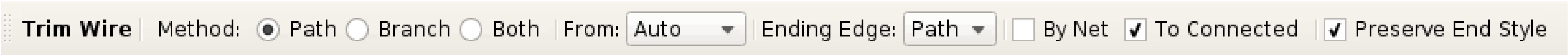


Width=12u

Stripe Width = 3 um
Number of Stripes = 3
Stripe Spacing = 1 um
Slot Length = 5 um
Slot Spacing = 2 um

Trim Wire Command

- **Used to trim interconnect wiring according to the selected method**
  - Path - operates sequentially on 1 or more Path/PathSeg to shorten or lengthen the object to connected neighbor
  - Branch - operates on 1 or more connected branches to prune away dangling objects
  - Both - Combination of Path and Branch methods

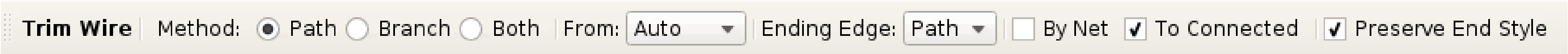


Trim Wire | Method: ● Path ○ Branch ○ Both | From: Auto ▾ | Ending Edge: Path ▾ | □ By Net ☑ To Connected ☑ Preserve End Style

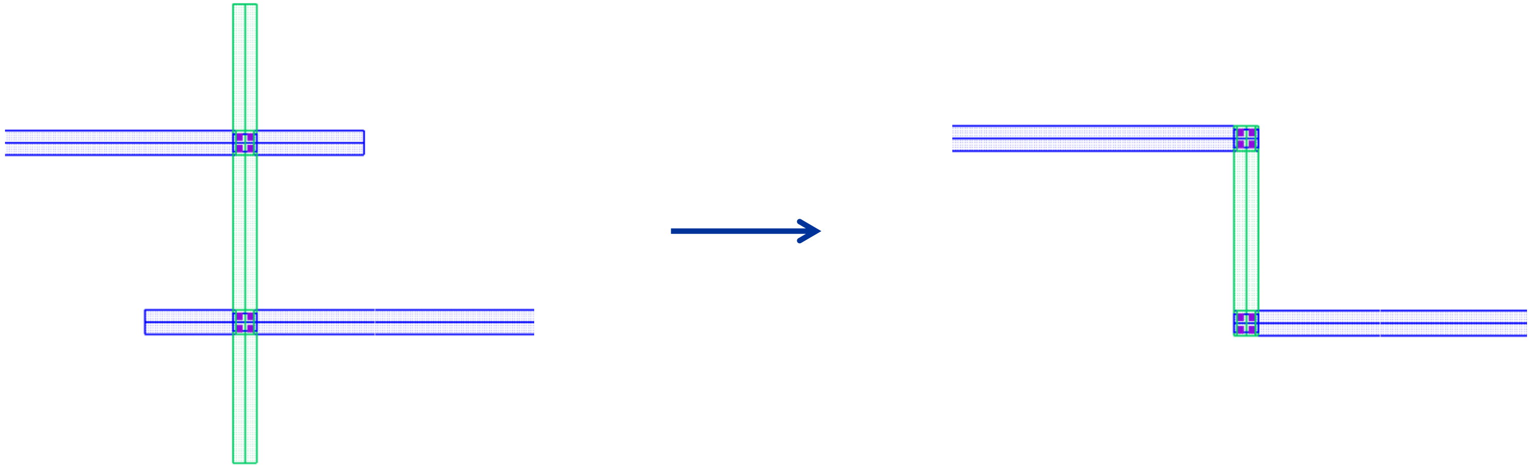
- **Can be invoked from Edit → Trim Wire or by using 7 bin key**

Trim Wire: Path Method

- Operate on a selection of figures, collection of figures, or figures on a net
- Control which edges to adjust
- Control which edge of connected neighbor to trim to
- Preserve begin and end style (PathSeg) and style(Path)

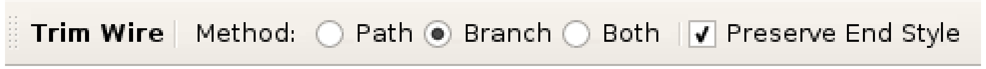


Trim Wire | Method: ● Path ○ Branch ○ Both | From: Auto ▾ | Ending Edge: Path ▾ | □ By Net ☑ To Connected | ☑ Preserve End Style

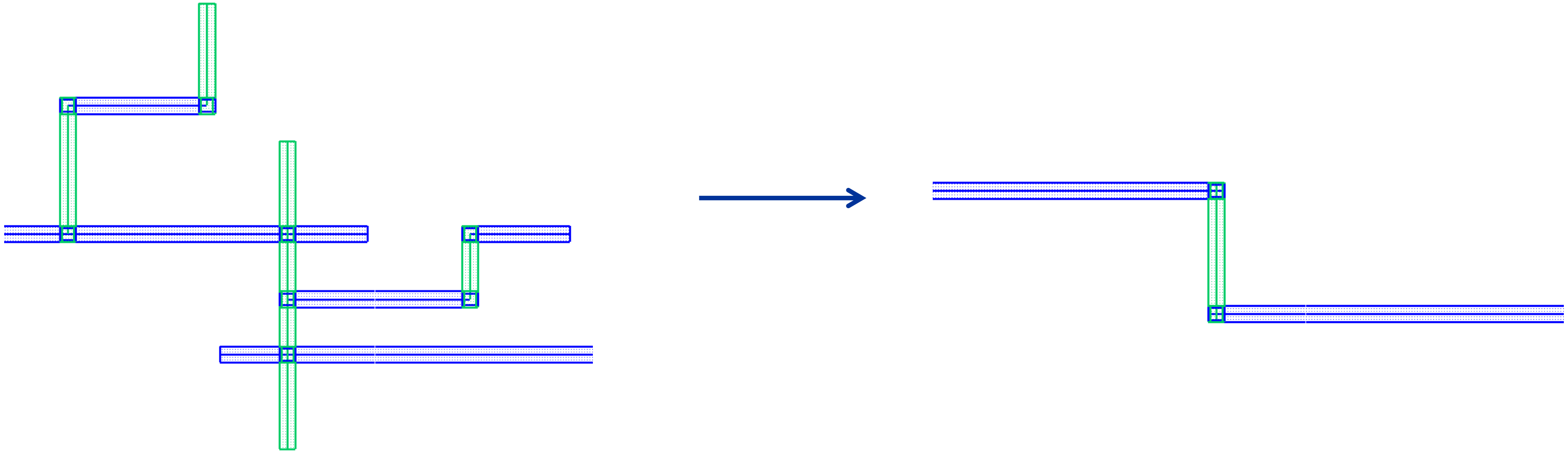


Trim Wire: Branch Method

- Operate on one or more branches by selecting objects belonging to the branch, or all branches on a net
- Preserve begin and end style (PathSeg) and style (Path)
- Removes entire branch (all figures on branch) if the branch is dangling or floating



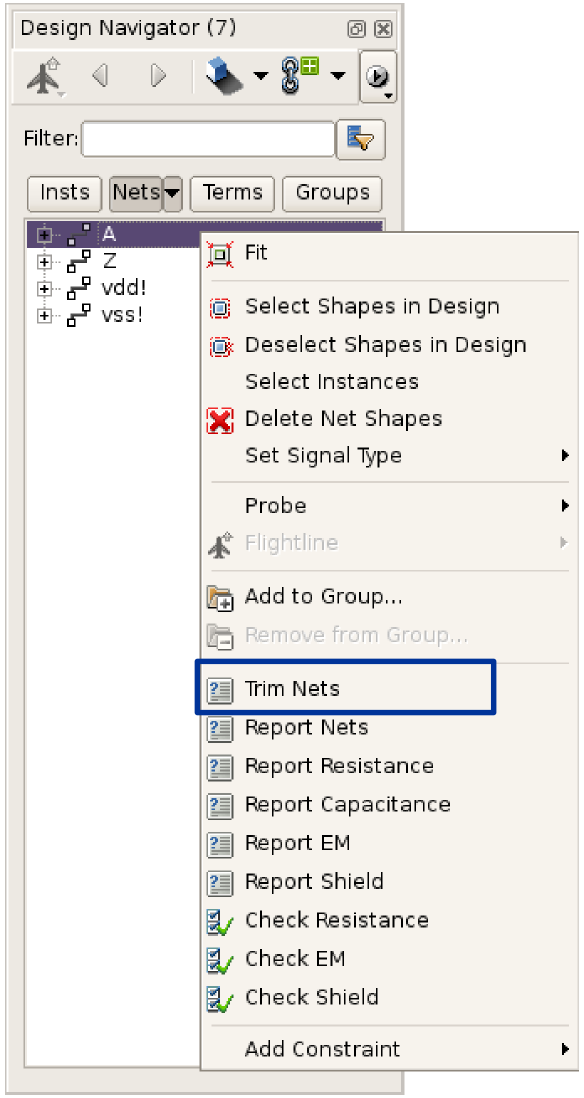
Trim Wire | Method: ○ Path ● Branch ○ Both | ☑ Preserve End Style



Trim Wire: Both Methods

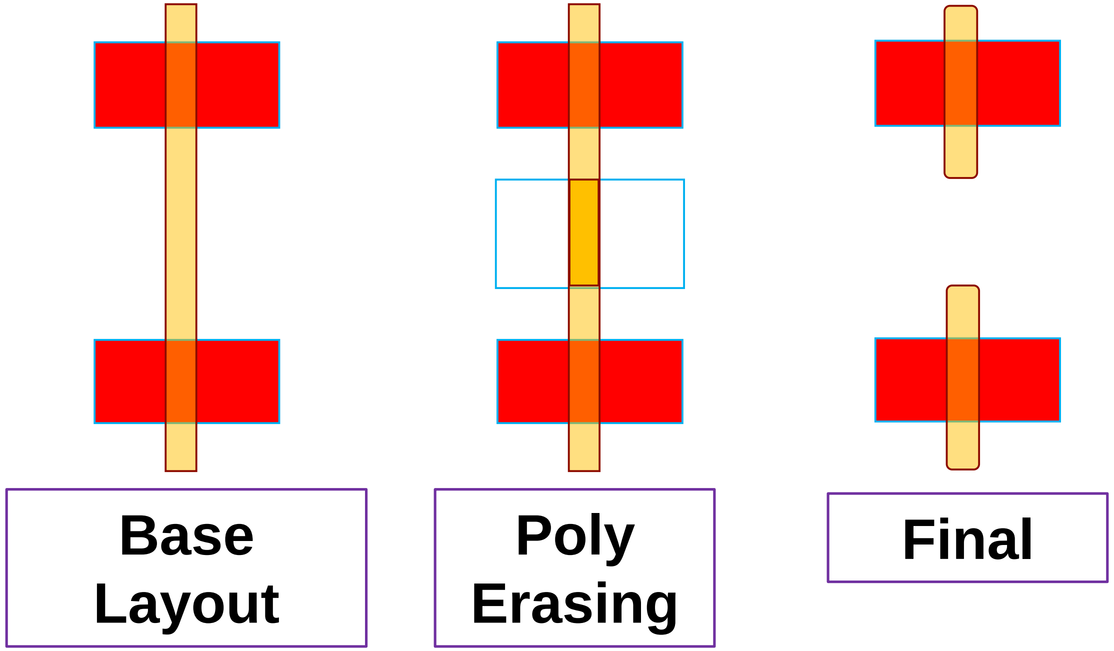
- **Combination of Path and Branch methods**
- **Will be used when executed on a net from Design Navigator or batch command**

```
le::trimWire $oaNet \
             -method both \
             -preserveEndStyle 1
```



Erase Layer

- **A special layer is introduced to improve poly shapes on Si**
  - Designers draw single line for stacked devices
  - Special marker will remove unnecessary poly during process
- **Custom Compiler provides a command to erase the poly (or other shapes) to keep nets separate**

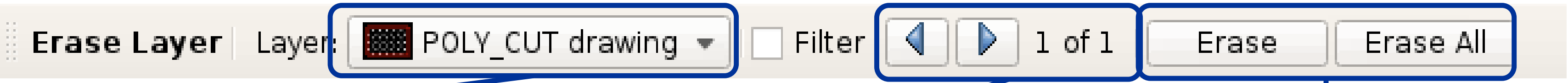


Erase Layer Command Use Model

- **Prerequisites in tech file**
- **erasingLayer constraint for defining the “cutting” and “cuttee” layers**
  - Tech file section example:

```
        (constraintGroups
                ( foundry
                …
                erasingLayer CutPoly (poly poly18)
                …
                )
        )
```

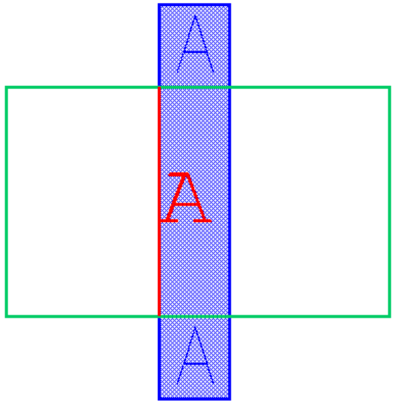
- **Call the ile::eraseLayer command from menu Edit -> Other -> Erase Layer**
  - Will erase “cuttee” shapes at 0x to separate nets and keep SDL compliancy



CutPoly layer

Apply cut to current cut marker or all

Navigate through cut marker shapes at 0x



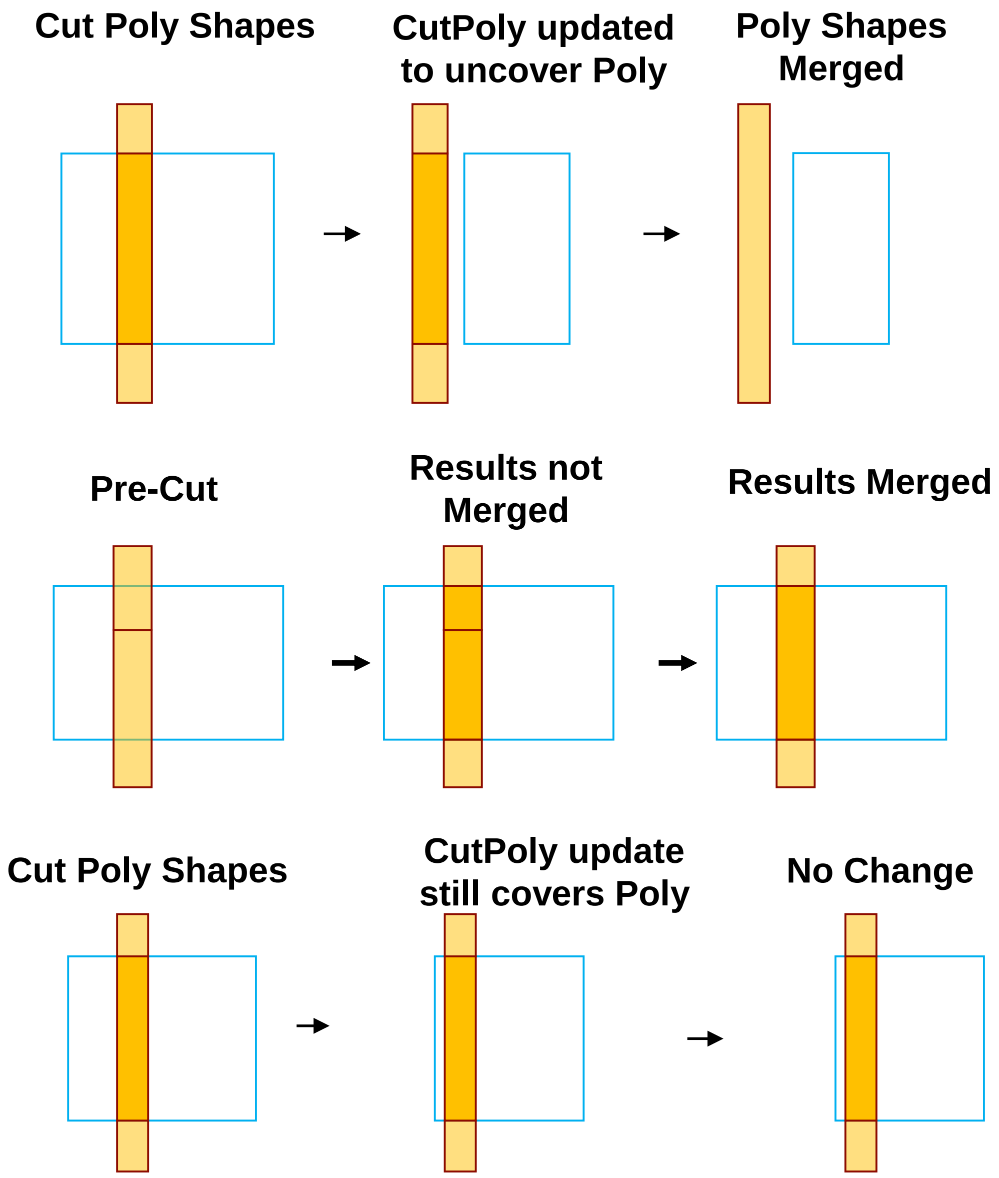
Results: Custom Compiler sees two nets
Middle segment not propagating

Examples of Edits After Erasing Poly Shapes

- **If CutPoly marker is moved**
  - Poly shapes are merged automatically

- **Poly shapes are merged within the CutPoly marker while cutting**

- **As long as the marker covers the cut poly shape, it remains as is**



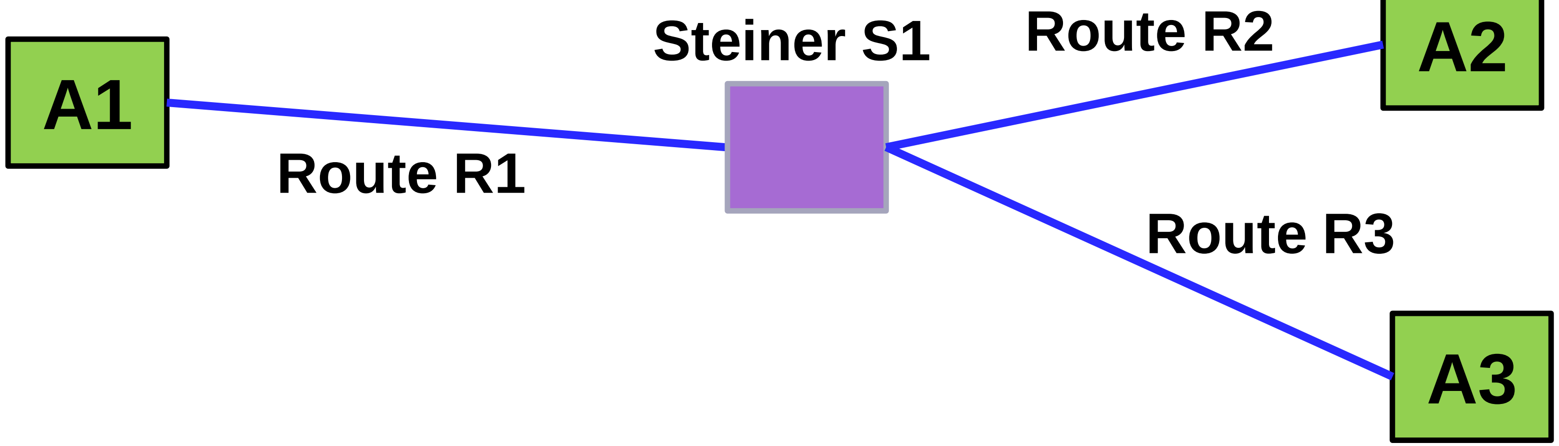
Cut Poly Shapes → CutPoly updated to uncover Poly → Poly Shapes Merged

Pre-Cut → Results not Merged → Results Merged

Cut Poly Shapes → CutPoly update still covers Poly → No Change

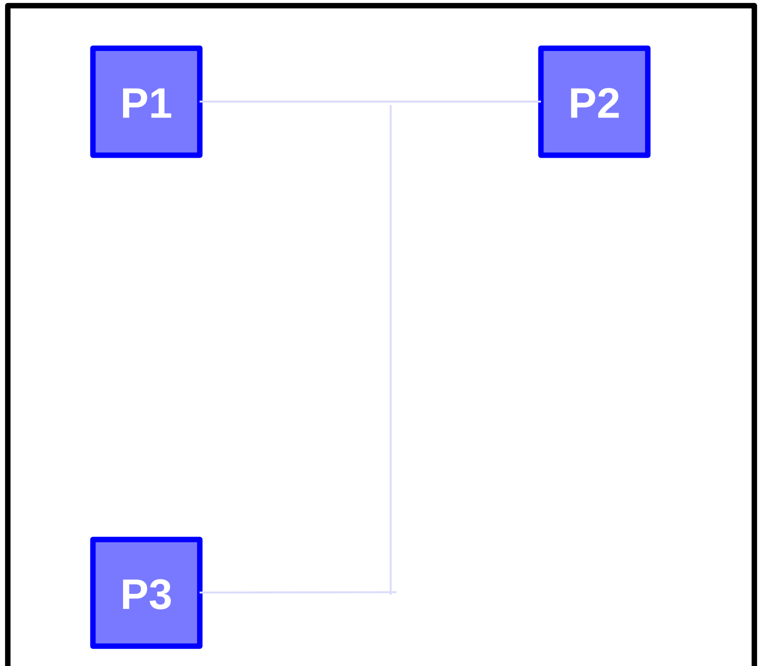
Topology

- **What is it?**
  - Allows the user to provide virtual pins or connection points for a net that divides the net into virtual net segments
    - oaSteiner == virtual pin/connection point
    - oaRoute == virtual net segment
  - Used to control the topology or location of routing
  - User can define constraints on individual oaRoutes
  - Per-segment control
    - Width
    - Layer
    - Spacing

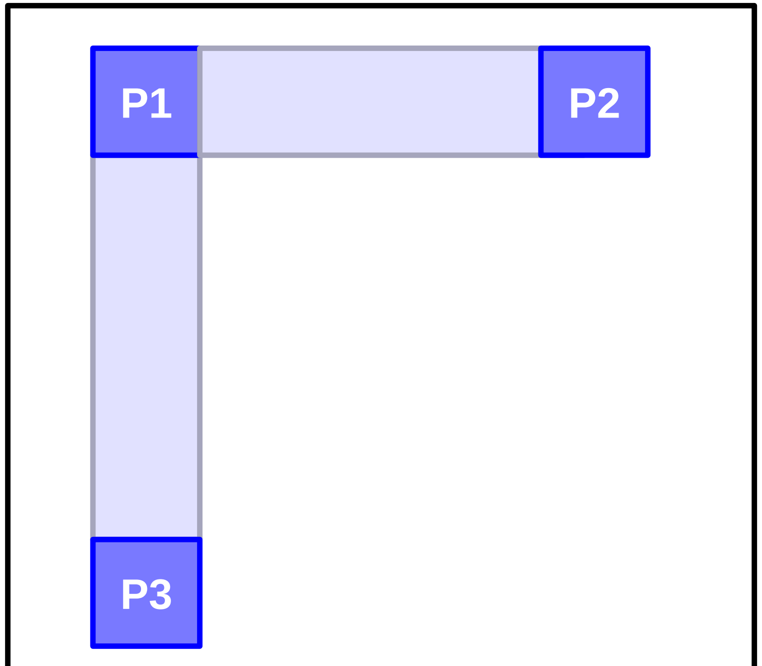


Topology Creation

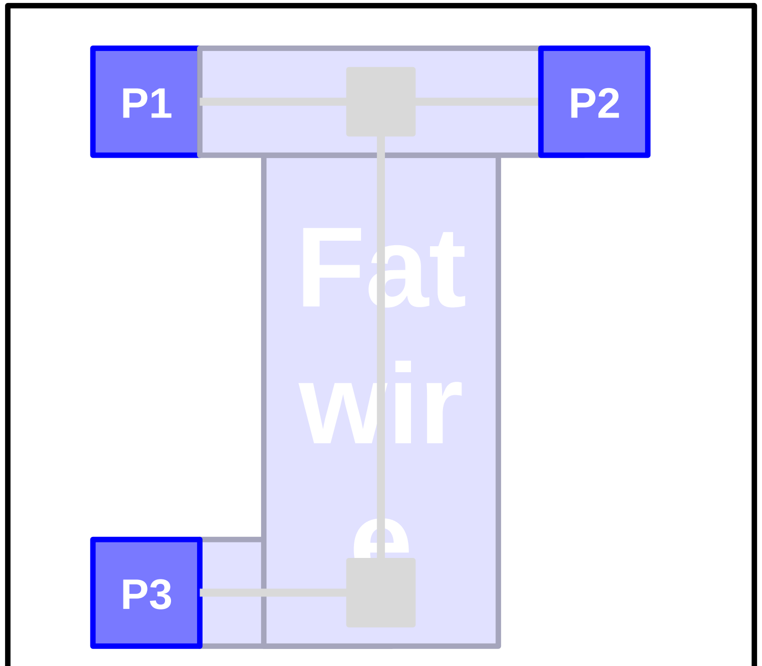
- Usage example



Unrouted
Desired topology



AutoRouter results

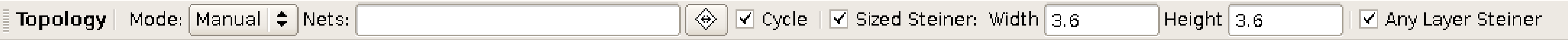


AutoRouter results with topology

Topology

- **Use Model**
  - Invoked from
    - Select Net from Design Navigator
    - Use Create → Topology
    - ile::createTopology TCL command

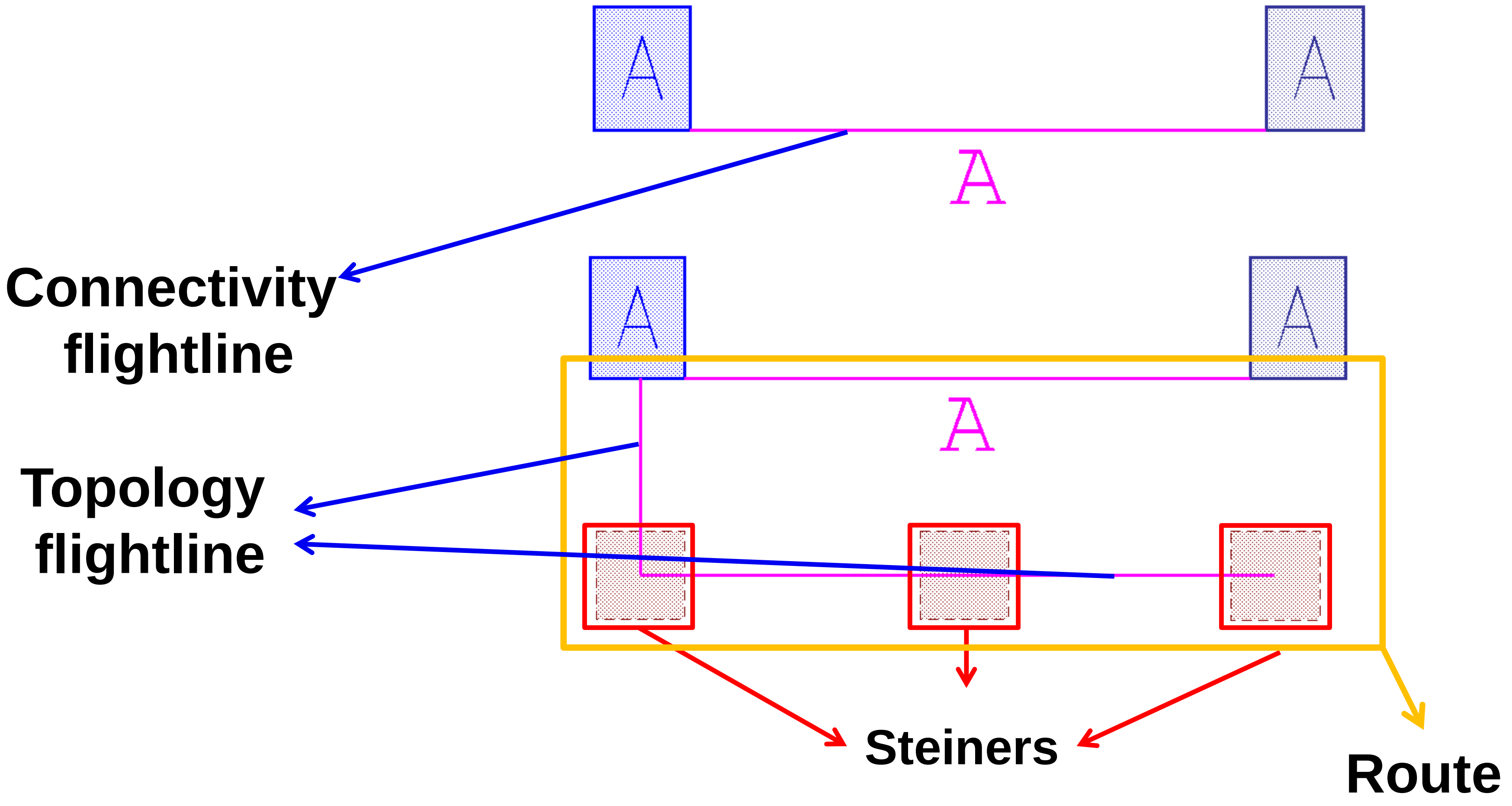
    - Create topology toolbar provides the following options:
      - Mode: Defines operation modes for creating topology
      - Nets: Defines net name(s) to be associated with topology objects to be created
      - Cycle: Defines whether the first net name in the net name field is removed or remains after object is committed
      - Sized Steiner: Steiner’s bBox size
        - Width/Height: Width/Height of Steiners
      - Any Layer Steiner: Defines whether create Steiner objects on any layer or use active layer from OLP
  - Basic point-click from source to destination to create Steiners
    - Can snap topology to the closest valid object using "s" bindkey
    - Can have intermediate points



Topology | Mode: Manual | Nets: | ☑ Cycle | ☑ Sized Steiner: Width 3.6 Height 3.6 | ☑ Any Layer Steiner

Topology

- **Example**



Connectivity flightline

Topology flightline

Steiners

Route

Topology

- **Preferences**

Preference Name	Type	Scope	Default	Description
leTopologyMode	enTopologyMode	cellview	manual	Operation mode for creating topology
leNetName	String	cellview	“”	Net name for ile::createTopology command
leCycleName	bool	cellview	True	Flag that specifies if we need to keep net name for subsequent objects that will be created in the same command
leSizedSteiner	bool	cellview	True	Flag that whether use specified steiner size or user interactively define it.
leSteinerWidth	float	cellview	0.1	Width of newly created steiners
leSteinerHeight	float	cellview	0.1	Height of newly created steiners
leAnyLayerSteiner	<bool	cellview	True	Flags that whether create steiner objects of any layer

Topology

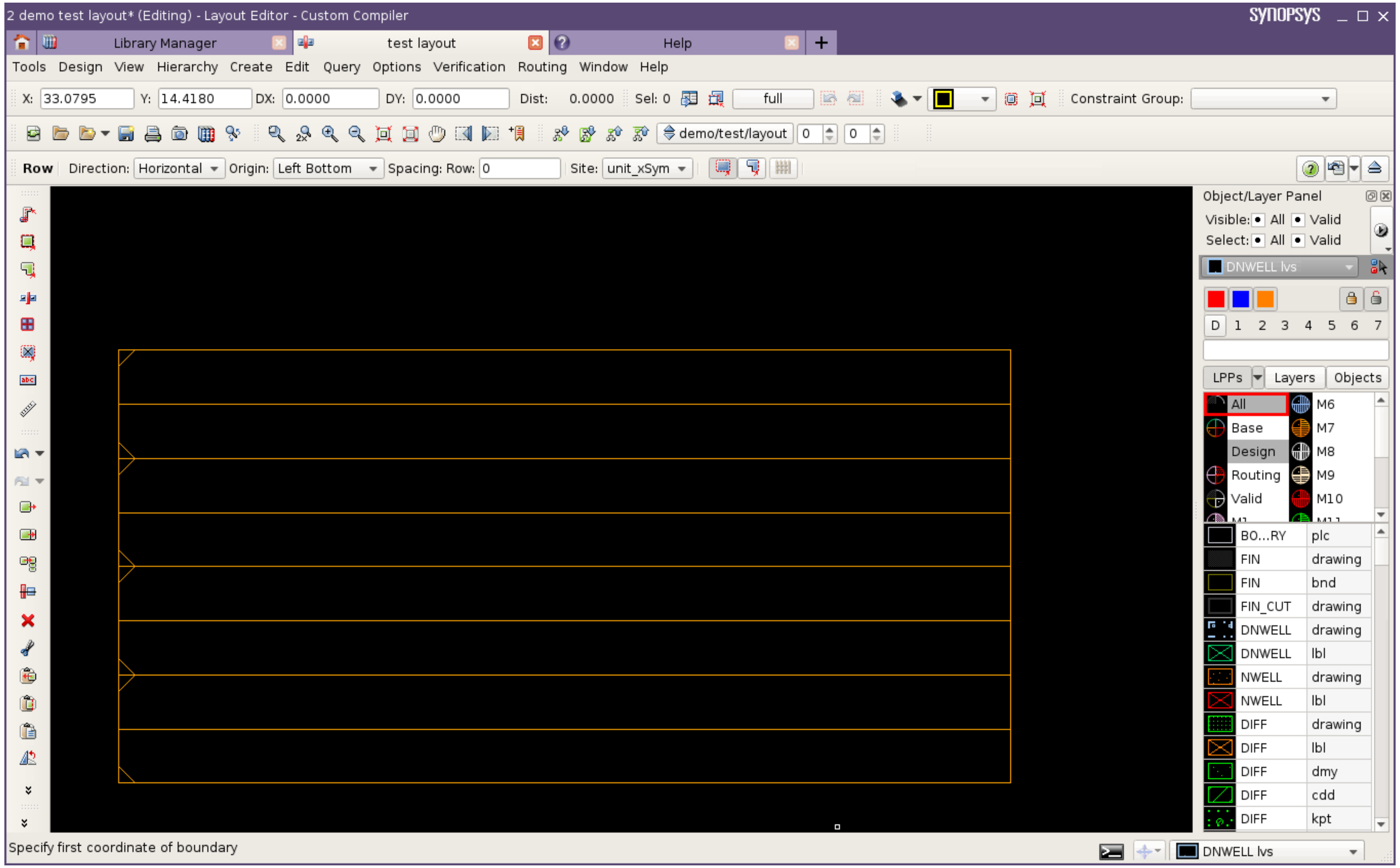
- **Topology vs. Connectivity Flightlines**
  - Connectivity Flightlines
    - Generated between same-net shapes
    - Not physically connected
    - Determined by the Connectivity Engine.
  - Topology flightlines
    - Generated on net topology
    - Based on oaRoutes
  - Both kinds of flightlines:
    - Can be static or dynamic
    - Assigned a color
    - Turned ON/OFF the same way
    - Only shown for the editable designs in an LE

Topology

Support both Steiner and Routes	Support Steiner Ignores Routes	Support only Steiner
Delete Attach/detach Group/ungroup Make cellview from selection Flatten	Split Reshape Merge Bridge interconnect Convert to polygon Create pins from selection	Move Stretch Copy Align Resize Copy to clipboard Cut Paste Chop Yank Rotate

Create Row

- **The Rows can be created in Custom Compiler**
  - Create → Row
  - Uses site definitions from technology



2 demo test layout* (Editing) - Layout Editor - Custom Compiler

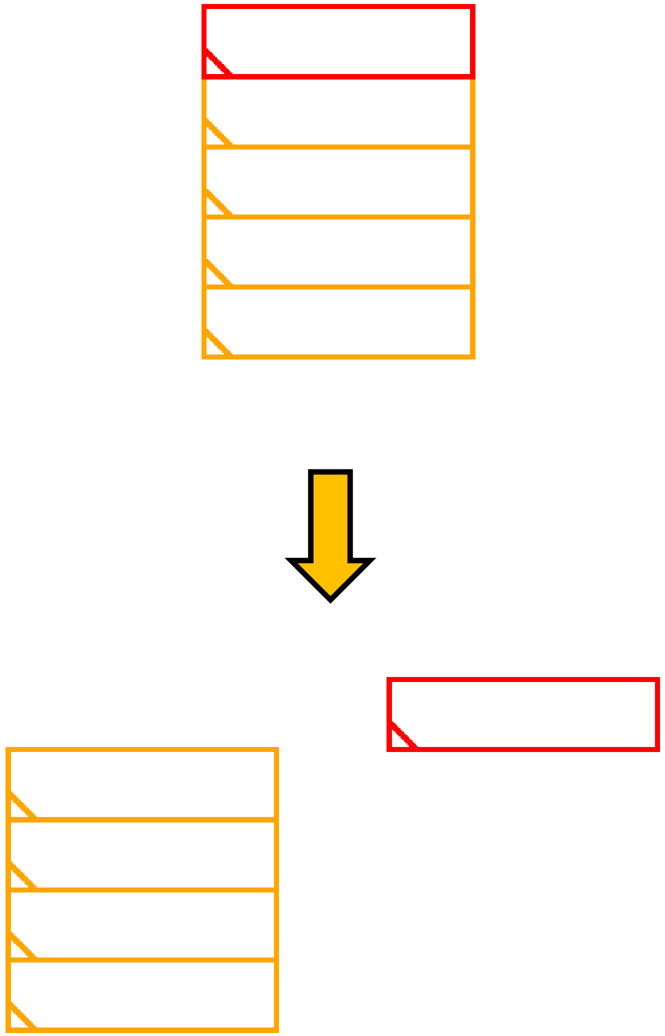
X: 33.0795 Y: 14.4180 DX: 0.0000 DY: 0.0000 Dist: 0.0000 Sel: 0

Row Direction: Horizontal Origin: Left Bottom Spacing: Row: 0 Site: unit_xSym

Specify first coordinate of boundary

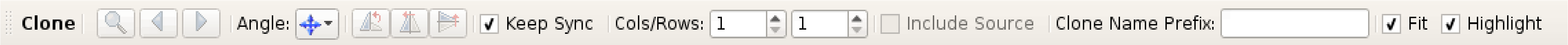
Create Row

- **Rows can be edited manually by using Move command:**
  - Supported Angle modes:
    - Orthogonal
    - Diagonal
    - Any Angle
  - Supported Rotate modes:
    - Clockwise
    - Flip Horizontally
    - Flip Vertically
- **Row Sites**
  - Number of sites can be modified in the Property Editor



Cloning

- **Edit → Instance → Clone command in the Layout Editor**



Clone | Angle: | ☑ Keep Sync | Cols/Rows: 1 1 | □ Include Source | Clone Name Prefix: | ☑ Fit | ☑ Highlight

- **Searches for clone patterns**
  - In Layout (e.g. ICC view) or Schematic if SDL is enabled
  - Source selection must include instances
    - Interconnect can be part of selection
- **Moves clone targets on cursor for placement**
  - Orientation of clones can be changed
  - Clones with Standard Cells will snap to row in ICC designs
- **Arrows allow to skip a clone target**
- **Already placed clones will not be in search results**

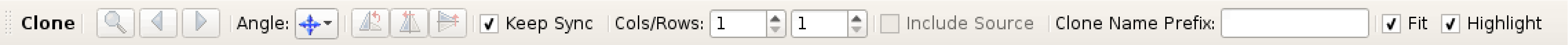
Finding Clones With TCL

lx::findCloneCandidates

- Returns a collection of objects from the layout view
- Input is a given collection of layout objects
- Command works when lxCloneFrom preference is set to layout or ICC layout
- The source clone objects must include instances

Synchronous cloning

- **Enabled with Keep Sync check box in Clone**
  - Prior to placing the clone targets

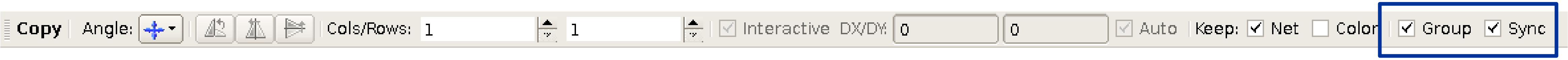


☑ Keep Sync Cols/Rows: 1 1 □ Include Source Clone Name Prefix: ☑ Fit ☑ Highlight

- **Clone targets are synchronized on placement**
  - All clones belong to same Synchronous Group
  - Any modification to one clone is replicated to all clones within the same Synchronous Group

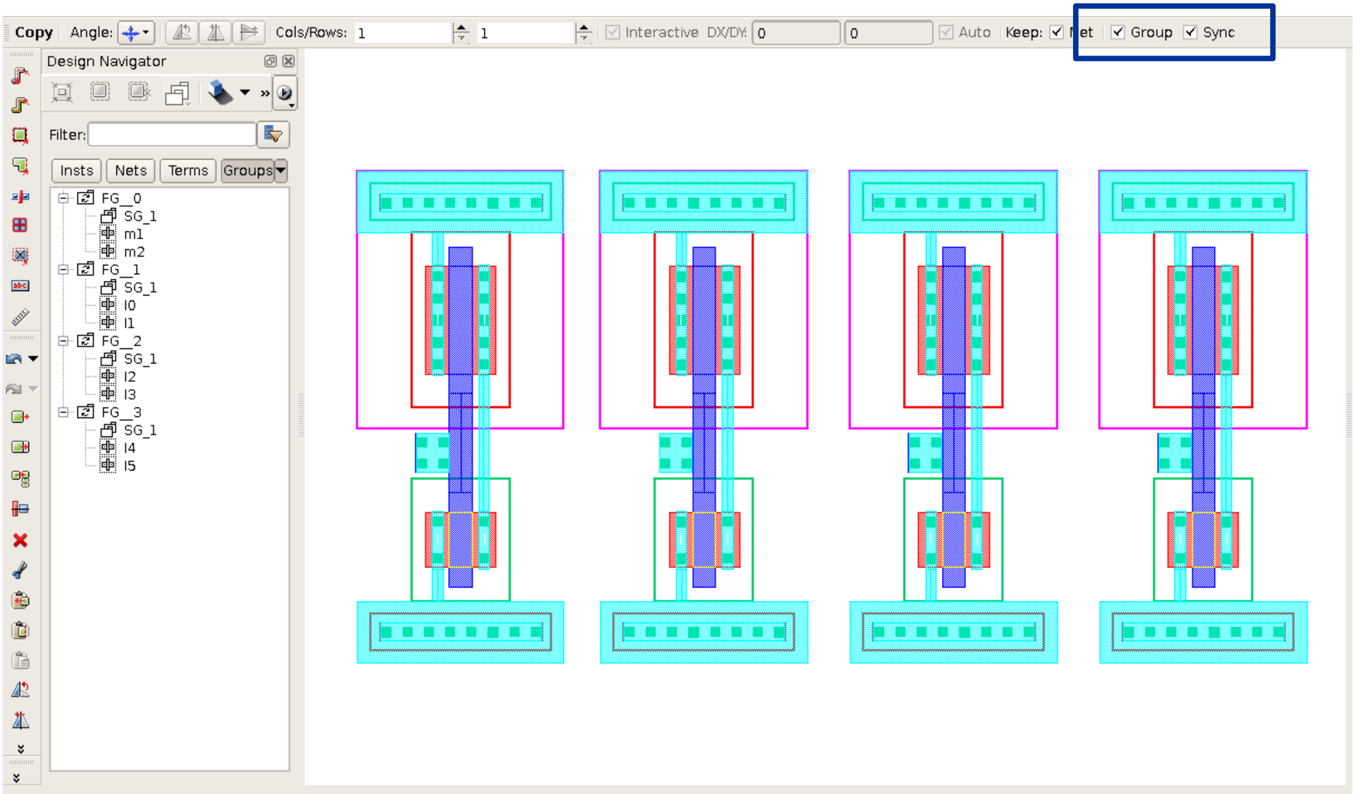
Object grouping during copy

- **Possibility to group objects that will be copied**



Copy | Angle: | Cols/Rows: 1 | 1 | ☑ Interactive DX/DY: 0 | 0 | ☑ Auto | Keep: ☑ Net □ Color | ☑ Group ☑ Sync

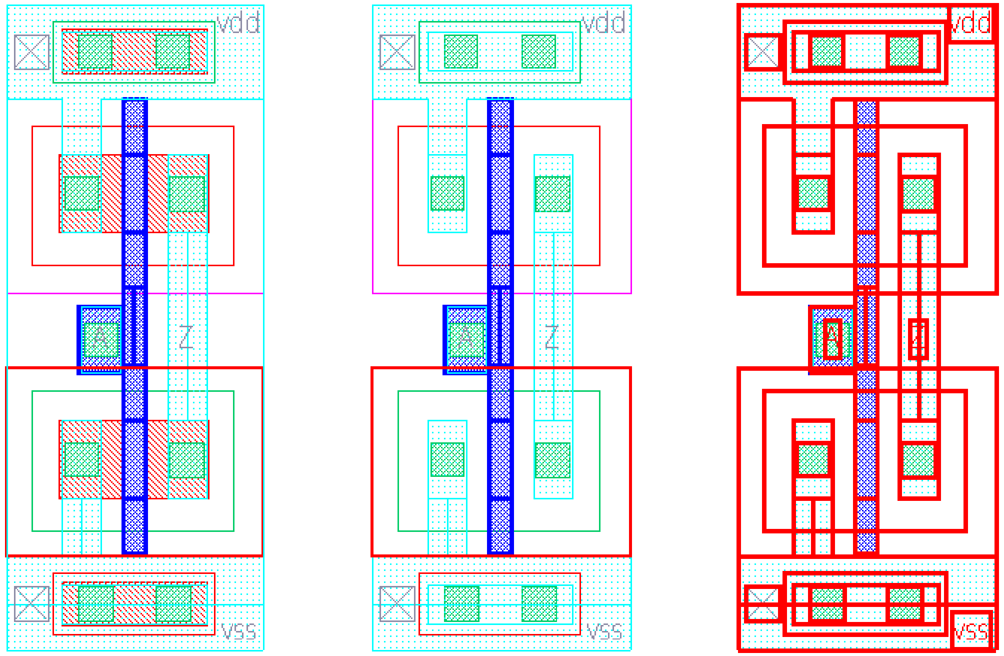
- **‘Keep Sync’ keeps copies as synchronous clones**
  - In ICC designs, operates only on interconnect
  - In OA layout, can apply to instances and interconnect



Copy | Angle: | Cols/Rows: 1 | 1 | ☑ Interactive DX/DY: 0 | 0 | ☑ Auto | Keep: ☑ Net | ☑ Group ☑ Sync

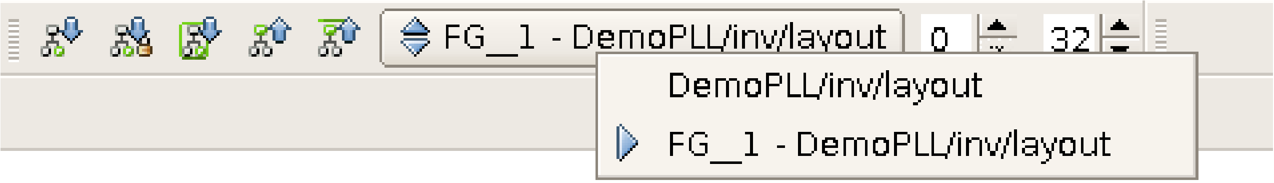
Repeat Copy Command

- **Invoked from Edit → Copy → Repeat Copy or by “.” bindkey**
- **Allows to repeat copy of last copied object or group of objects**
  - Keeps same distance used during last copy
  - Keeps same direction used during last copy



Edit In Group (EIG)

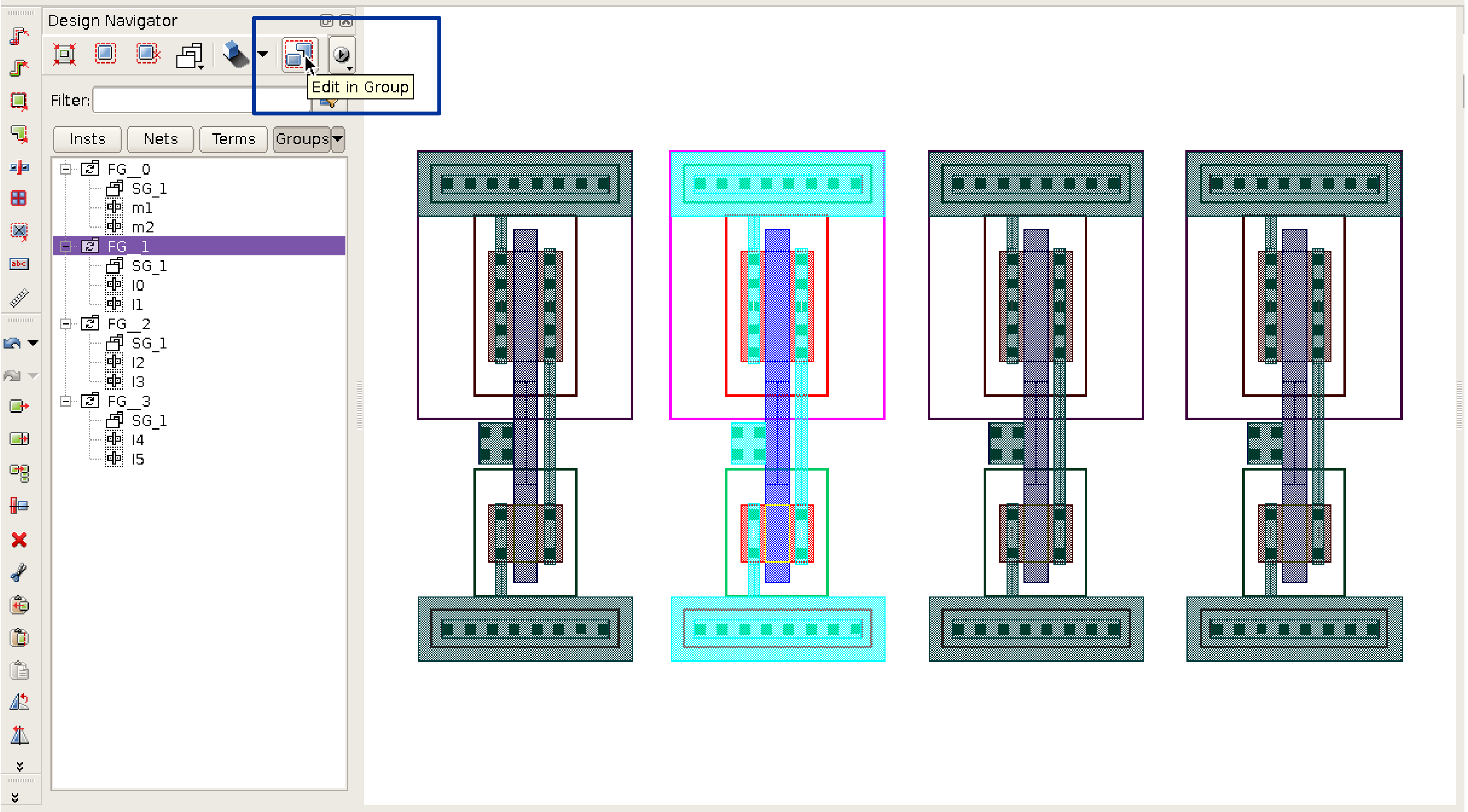
- **Edit In Group allows to edit figure groups created with Edit Clone**
  - All edits to any clone will be synchronized to others immediately
- **Use Hierarchy → Edit In Group**
  - Hierarchy → Return to return or Ctrl-E
- **The level being edited shown in the hierarchy pull down button**



- **To identify clearly what is being edited, use EIG shadow**

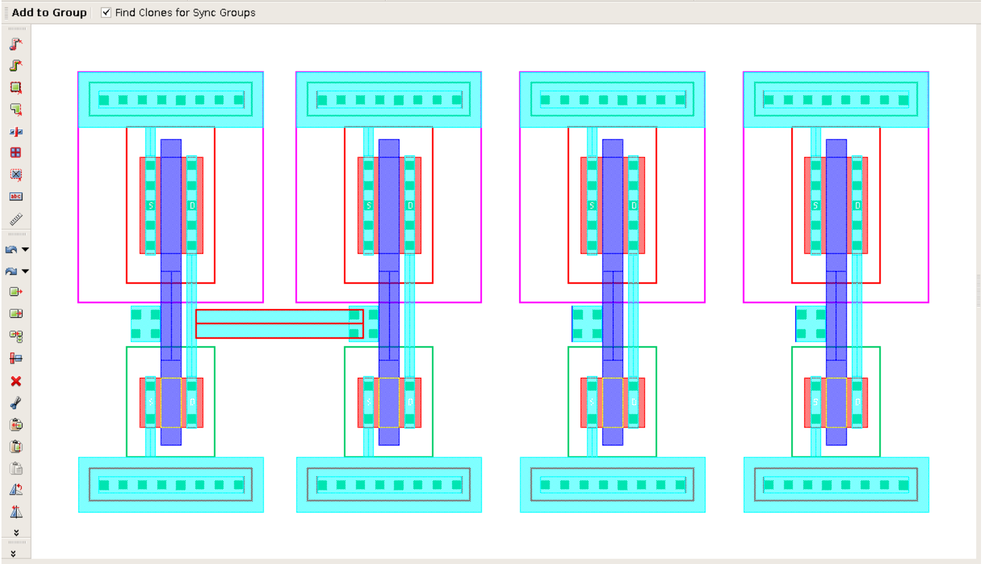
  db::setPrefValue deShadowEIG –value true

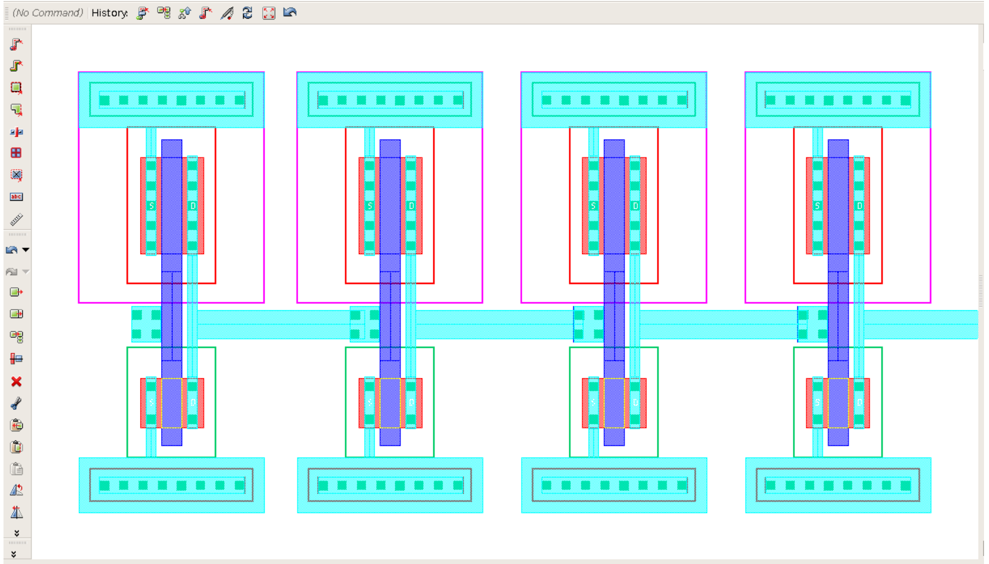
Edit In Group Example



Add To Group

- Allows designers to add an object at the top level to a any figure group or clone
- Call from Edit → Hierarchy → Add To Group
- Select all the shapes to add to the clone
- Select the clone
- All clones are updated



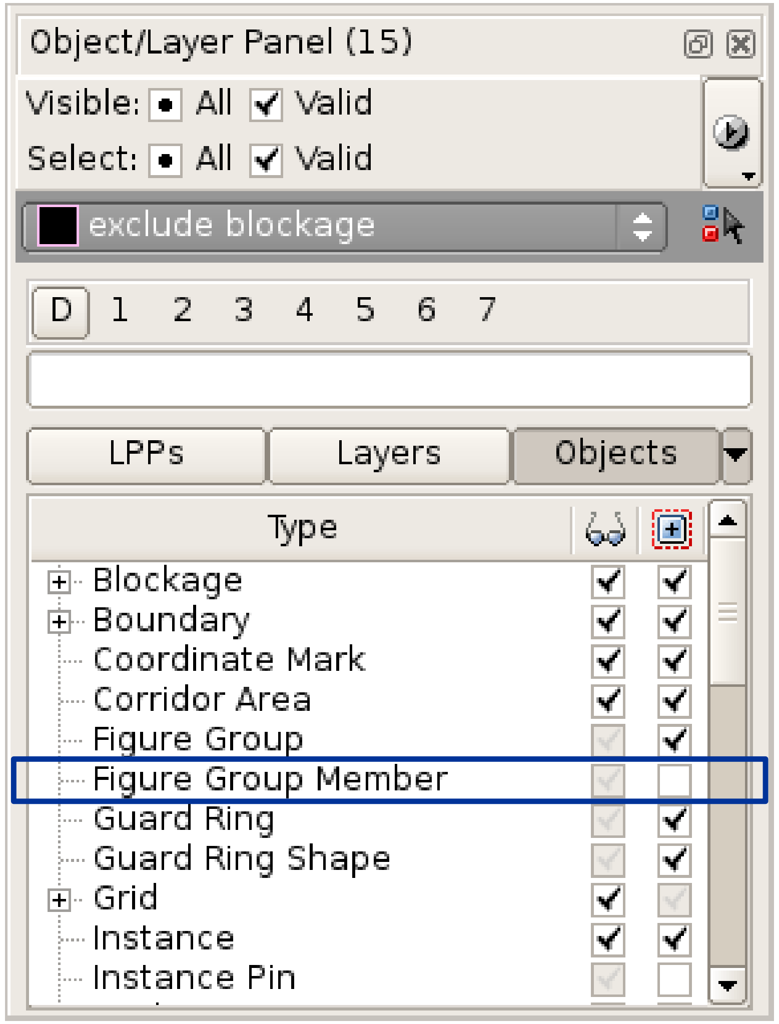


Remove From Group

- In case a route needs to be different for one clone, use Remove From Group
- First EIG in any clone of the Synchronous Group
- Call Edit → Hierarchy → Remove From Group
- Select the shapes to remove
- The shapes are moved to the top level for all synchronized clones

Editing Figure Group Members

- **Allows to edit figure groups members directly**
  - Option Figure Group Member should be enabled from
    OLP Object Panel
  - When enabled, allows to select Figure Group Member directly



Object/Layer Panel (15)

Visible: ● All ☑ Valid

Select: ● All ☑ Valid

exclude blockage

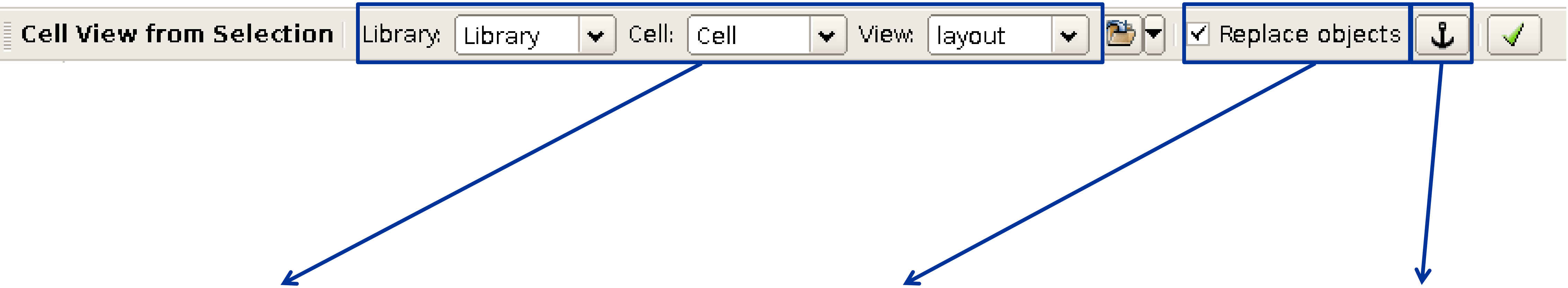
D 1 2 3 4 5 6 7

LPPs | Layers | Objects

Type		
Blockage	✓	✓
Boundary	✓	✓
Coordinate Mark	✓	✓
Corridor Area	✓	✓
Figure Group	✓	✓
Figure Group Member	✓	□
Guard Ring	✓	✓
Guard Ring Shape	✓	✓
Grid	✓	✓
Instance	✓	✓
Instance Pin	✓	□

Make CellView From Selection

- **Edit → Hierarchy → Make CellView**



Cell View from Selection

Library: Library

Cell: Cell

View: layout

☑ Replace objects

Newly created cellView

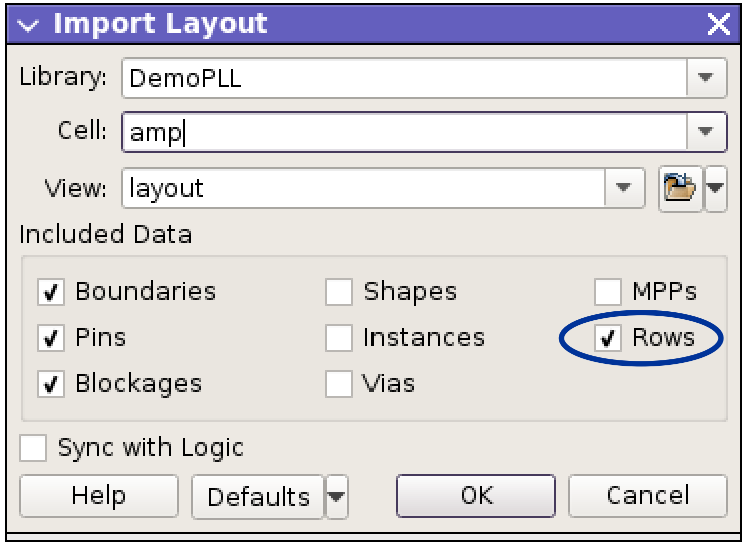
If true, will replace the selected objects by newly created cellView

For defining the origin of the new cellView

- Works only for fully selected objects

Import Layout

- **Create → Import Layout**
- **Import Layout can be used to import an ICC block to LE**
- **Option to import Rows**
  - Copy the Row information across cells
  - Importing Rows only works when importing layout between OA designs.
  - Does not work for designs opened through ICC plug-in



Library: DemoPLL

Cell: amp

View: layout

Included Data

☑ Boundaries □ Shapes □ MPPs

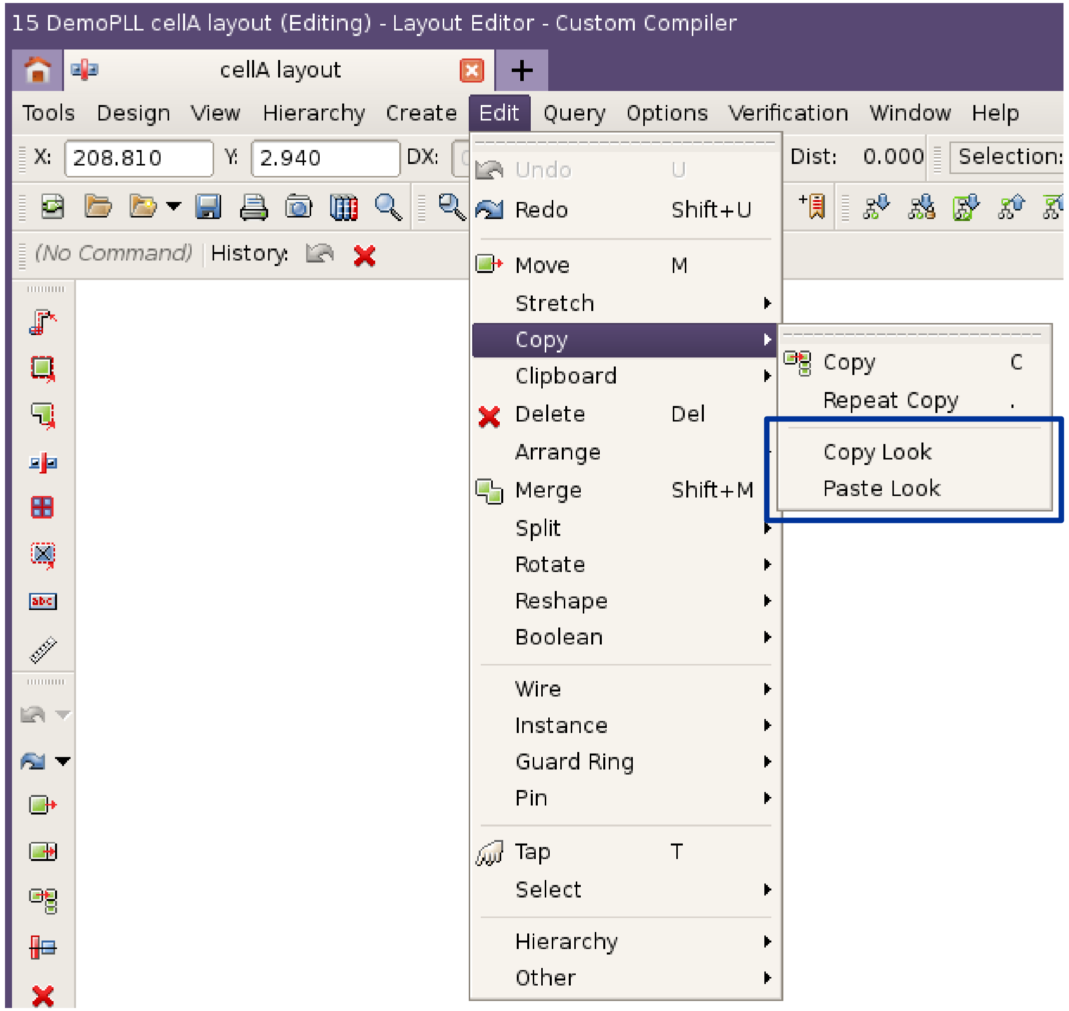
☑ Pins □ Instances ☑ Rows

☑ Blockages □ Vias

□ Sync with Logic

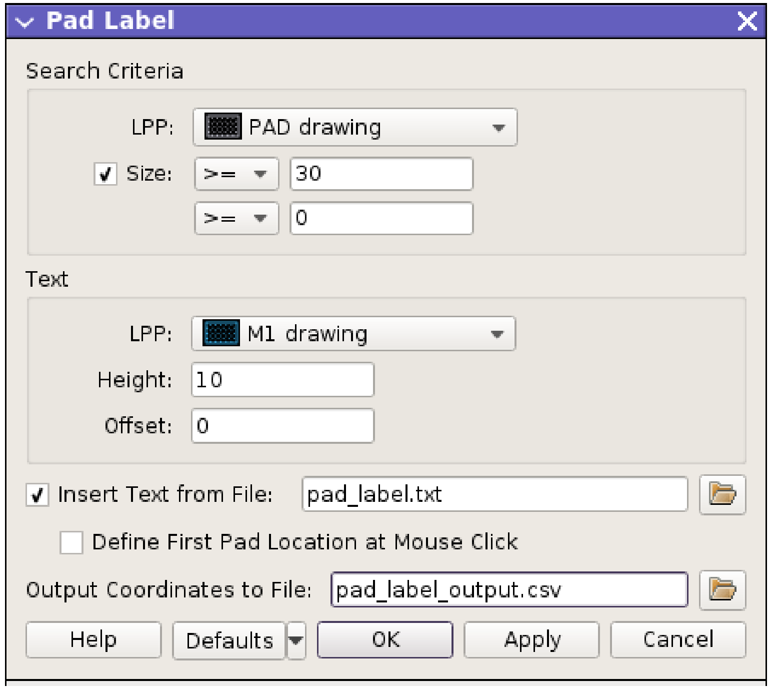
Copy Look and Paste Look

- **Allows user to copy and paste instance and via parameters from one object to another**
  - Parameters managed by SDL will be ignored during copy/paste look process



Pad Label

- **Pad Label Command creates labels on bound pad shapes**
  - Bound pads are defined based on size and LPP filtering
- **Edit → Other → Pad Label**
- **Create label**
  - Create label on specified LPP
  - Text of the label is from a file
    - Syntax follows Create Label with Expand
      - A B C D E
      - 1:5 → 1, 2, 3, 4, 5
      - A<1:5> → A<1>, A<2>, … A<5>
      - A<1:10:3> → A<1>, A<4>, A<7>, A<10>
- **Can dump pad label location to a CSV file with the following format:**
  - Label Name, X-coordinate of label, Y-coordinate of label, width of pad, height of pad, BBox of pad



Pad Label

Search Criteria

LPP: PAD drawing

☑ Size: >= 30

>= 0

Text

LPP: M1 drawing

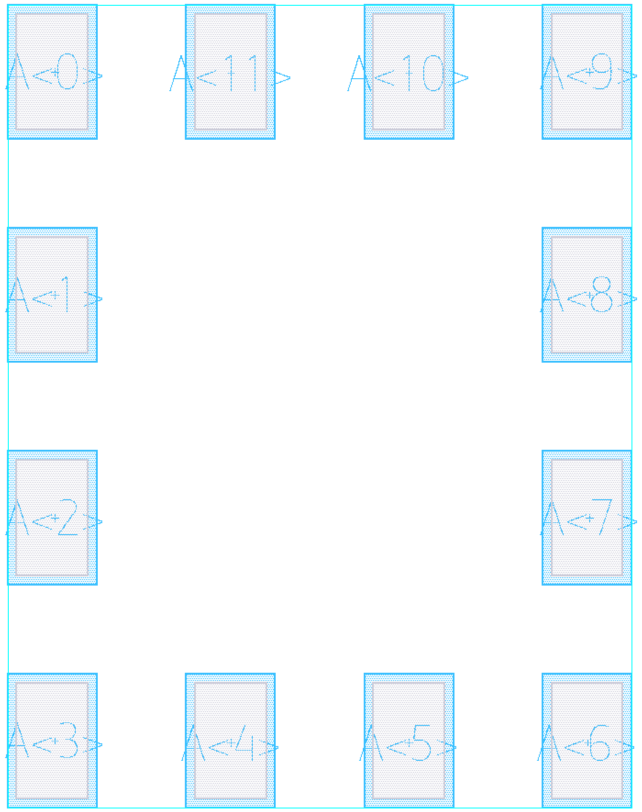
Height: 10

Offset: 0

☑ Insert Text from File: pad_label.txt

☐ Define First Pad Location at Mouse Click

Output Coordinates to File: pad_label_output.csv



```
ile::padLabel
```

```
le::padLabel
```

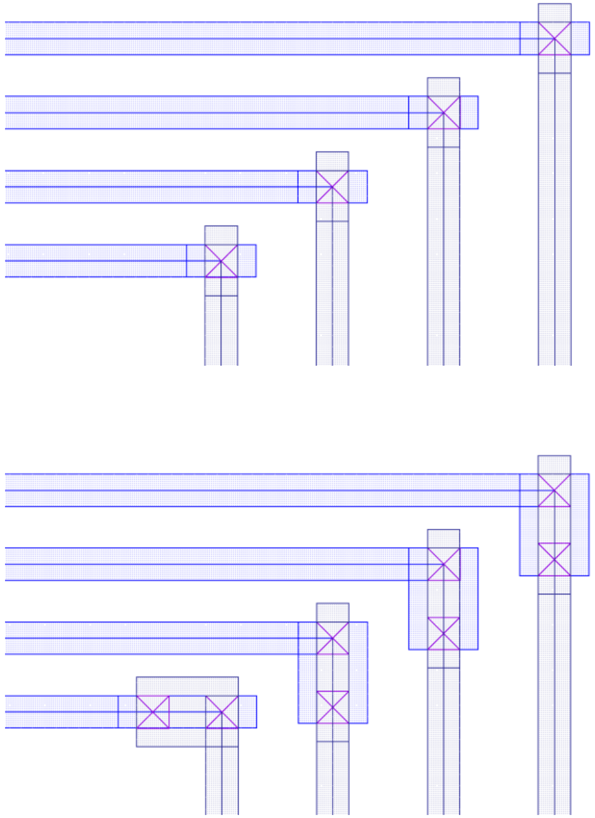
Replace Vias

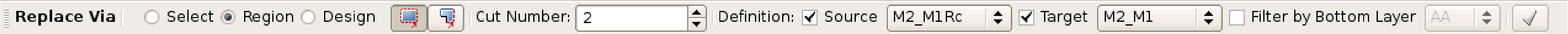
- **Allows to replace vias with a larger number of via cuts to improve reliability**
  - Located under **Query→ Replace Vias**
- **Interactive command**
  - Works on a selected via only
  - Can support changing N cuts to M cuts
  - Prefers single row or column
    - Array of via cuts possible
  - Grows vias along longer path first
    - Except in case of obstacles
  - Replacement By Region and Design
    - By Region: Click, Release and Drag to draw region
  - Source and Target via type to replace
- **Preference**

  `leReplaceViaCutNum` [Default: 2]
- **Command name**

  ```
  ile::replaceVia
  ```
- **Non-interactive command**

  ```
  le::replaceVia -design <oaDesign>
  [-sourceCutNum <int>] [-targetCutNum <int>]
  ```

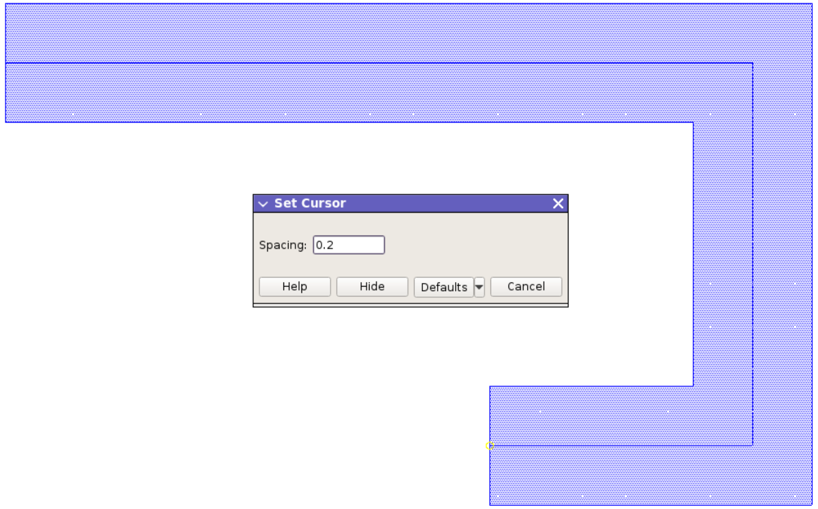




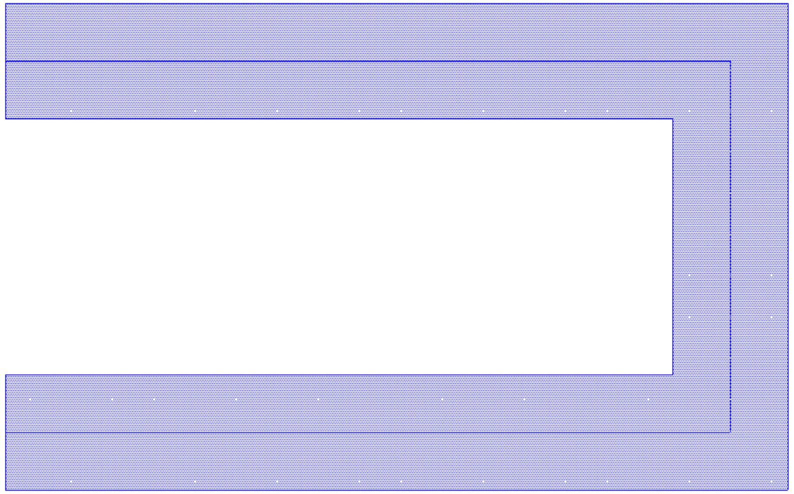
Replace Via ○ Select ● Region ○ Design Cut Number: 2 Definition: ☑ Source M2_M1Rc ☑ Target M2_M1 □ Filter by Bottom Layer AA ✓

Set Cursor

- **Allow to set cursor to the user defined coordinated during creation and editing**
  - Works with editing and creation commands
  - Use Shift-Space binding to activate command
  - Set the value and activate edge that need to be set
  - Allow to input negative values

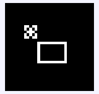
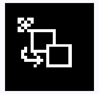
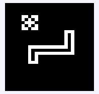
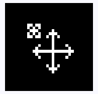
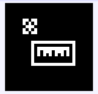
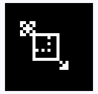
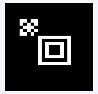


Spacing: 0.2



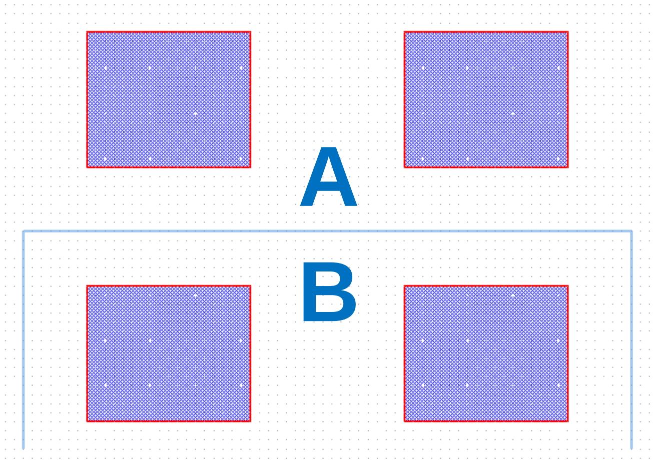
Context Sensitive Cursor

- **The cursor’s look changes depending on the command**
  - Helps the user know which command is active
- **Examples shown in table below**

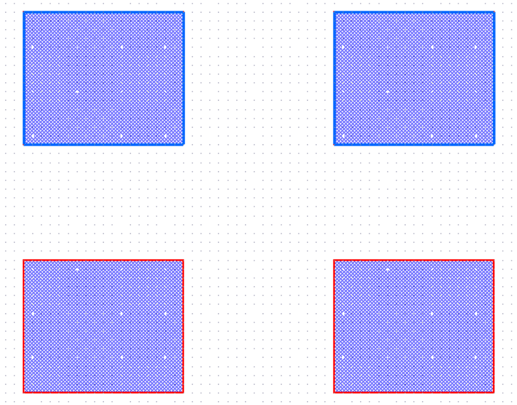
Command	**Cursor Icon**
Create Rectangle	
Copy	
Create Interconnect	
Move	
Ruler	
Stretch	
Create Via	

Line Select

- **Allows to split selection set to two separate sets with further toggling between sets**
  - Use Alt-V to call line select
  - Available within edit commands
  - Operates only on already selected objects
  - Use model:
    - Draw line to separate layout in two regions and commit by Enter
      - Use Backspace to backtrack the previous point
      - Use Spacebar to reset the line
    - Click in region to create a sub selection
      - Use Backspace to deselect only sub selection
      - Use Ctrl-' to deselect both sub and original selections
    - Complete the edits on the sub selection



Click A region to select

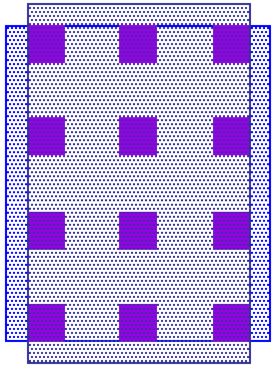


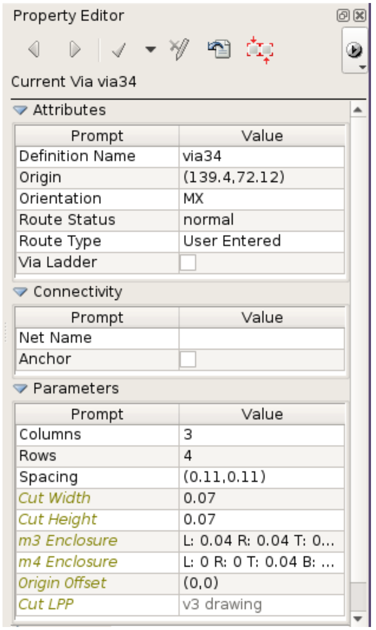
Ready to edit

Built-In Custom Vias

- **OA custom vias with built-in procedures for selection, stretch, chop**
  - Identical to standard vias but extends the capability of standard vias especially in chop functionality
  - Fully integrated with creation and editing commands
  - Built-in custom vias represented just like standard vias but with more powerful functionalities
    - Can chop layer1 of built-in custom via to any shape while standard via only supports the rectangle type
    - When operated on a standard vias and the editing operation is not supported, standard vias will be implicitly changed to built-in custom vias to perform this editing

Standard Via





Property Editor

Current Via via34

Attributes

Prompt	Value
Definition Name	via34
Origin	(139.4,72.12)
Orientation	MX
Route Status	normal
Route Type	User Entered
Via Ladder	□

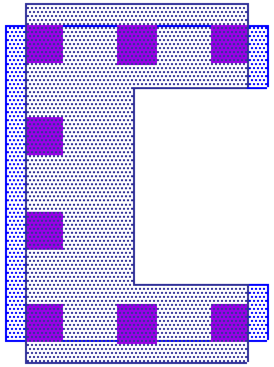
Connectivity

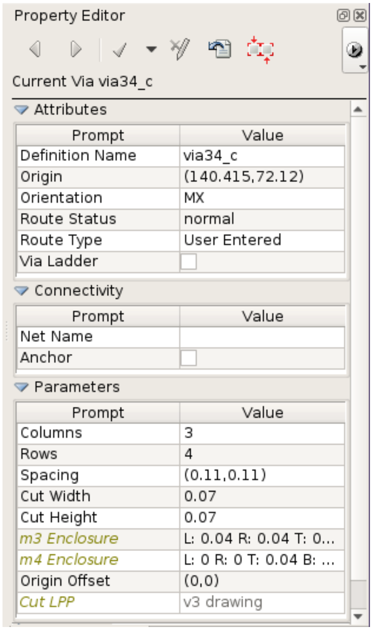
Prompt	Value
Net Name	
Anchor	□

Parameters

Prompt	Value
Columns	3
Rows	4
Spacing	(0.11,0.11)
Cut Width	0.07
Cut Height	0.07
m3 Enclosure	L: 0.04 R: 0.04 T: 0...
m4 Enclosure	L: 0 R: 0 T: 0.04 B: ...
Origin Offset	(0,0)
Cut LPP	v3 drawing

Built-In Custom Via





Property Editor

Current Via via34_c

Attributes

Prompt	Value
Definition Name	via34_c
Origin	(140.415,72.12)
Orientation	MX
Route Status	normal
Route Type	User Entered
Via Ladder	□

Connectivity

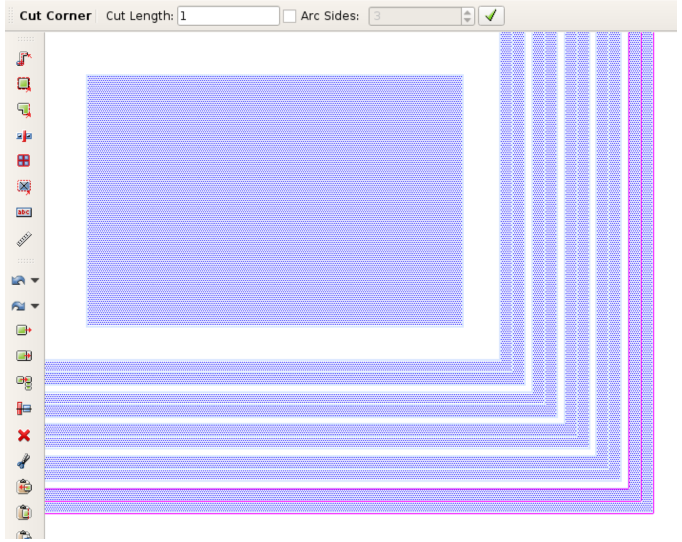
Prompt	Value
Net Name	
Anchor	□

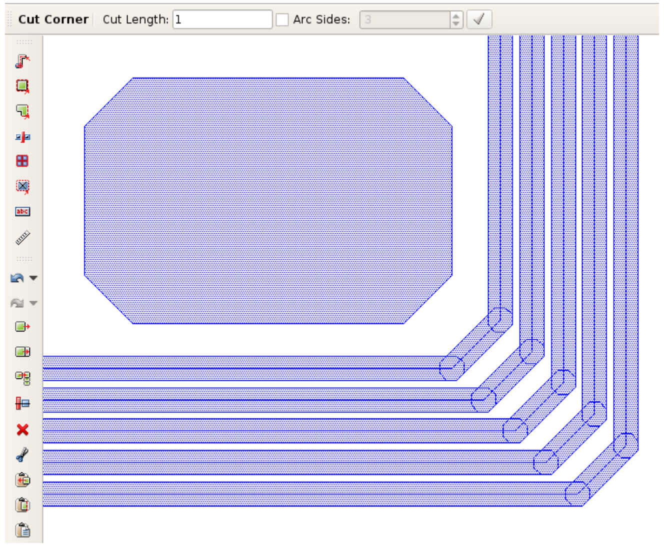
Parameters

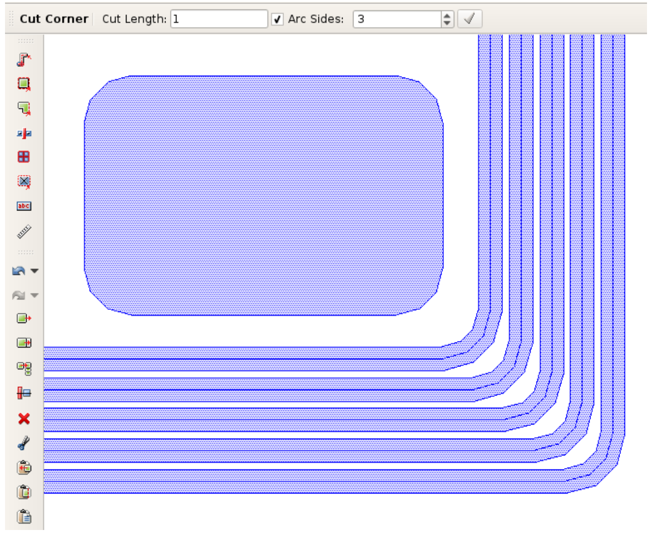
Prompt	Value
Columns	3
Rows	4
Spacing	(0.11,0.11)
Cut Width	0.07
Cut Height	0.07
m3 Enclosure	L: 0.04 R: 0.04 T: 0...
m4 Enclosure	L: 0 R: 0 T: 0.04 B: ...
Origin Offset	(0,0)
Cut LPP	v3 drawing

Cut Corner

- **Allows to cut corners of shapes and busses**
  - User can specify Cut Length for corner cut
  - Corners can be cut to Arcs
    - Arc Sides should be enabled
    - Number of sides can be specified
- **Edit > Other > Cut Corner (Alt-B)**

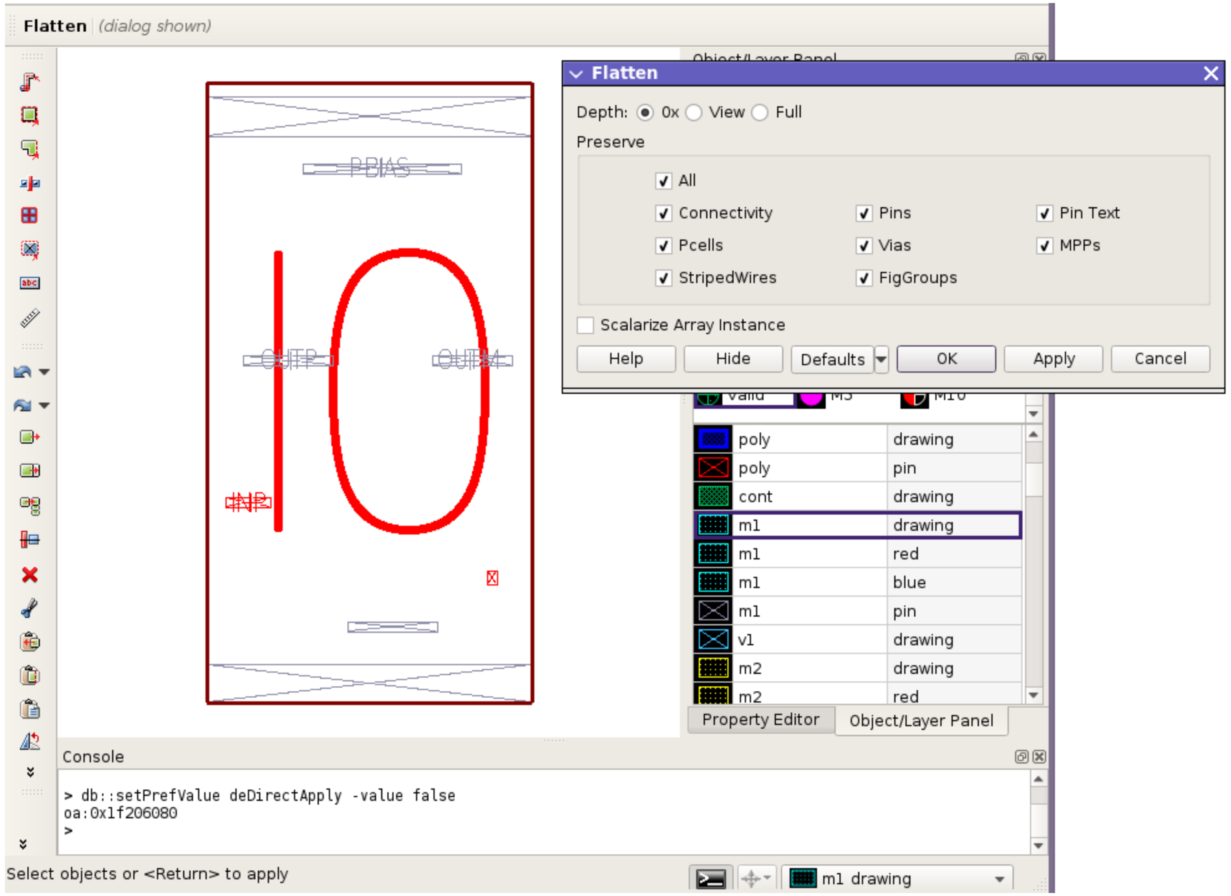

Cut Corner Cut Length: 1 □ Arc Sides: 3 ✓


Cut Corner Cut Length: 1 □ Arc Sides: 3 ✓


Cut Corner Cut Length: 1 ☑ Arc Sides: 3 ✓

Direct Apply Control for Editing Commands

- **Use preference deDirectApply**
  - Allows to control whether editing command will be directly applied in pre-selection mode or not
    - Particularly designed for Flatten command to allow users to control Flatten in pre-selection mode
  - Default value is false



Depth: ● 0x ○ View ○ Full

Preserve

☑ All

☑ Connectivity ☑ Pins ☑ Pin Text

☑ Pcells ☑ Vias ☑ MPPs

☑ StripedWires ☑ FigGroups

□ Scalarize Array Instance

poly	drawing
poly	pin
cont	drawing
m1	drawing
m1	red
m1	blue
m1	pin
v1	drawing
m2	drawing
m2	red

```
> db::setPrefValue deDirectApply -value false
oa:0x1f206080
>
```

Select objects or <Return> to apply

m1 drawing

```
db::setPrefValue deDirectApply -value 1
```

Test For Understanding

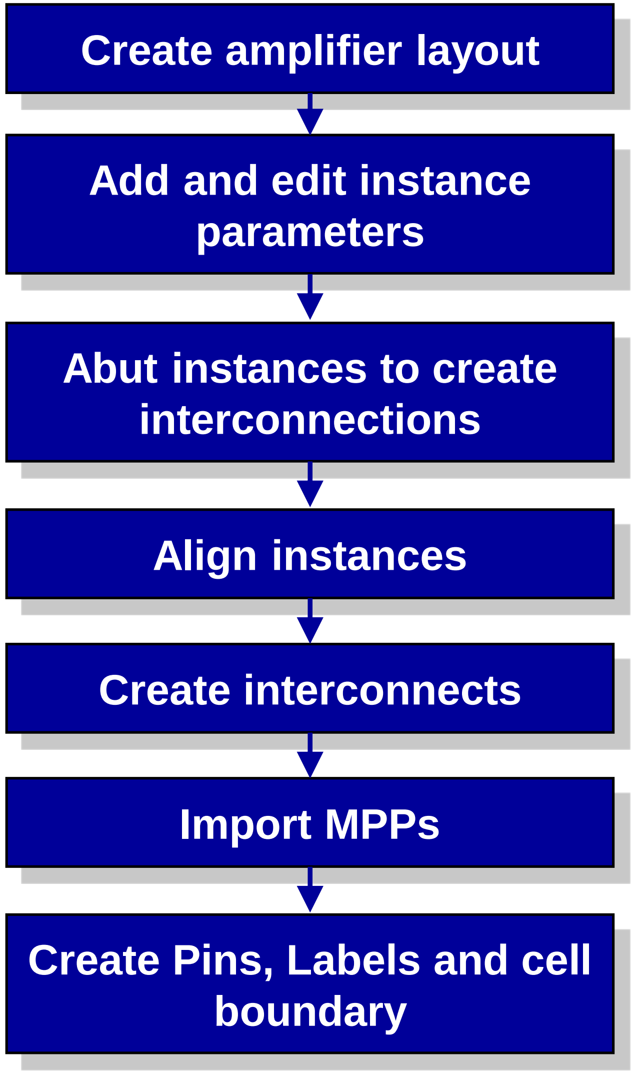
3 Minutes!

- Abutment adjusts placement of one instance with respect to another instance: True / False
- Split requires selection of editable objects: True / False
- Edit In Place can be used for editing Synchronous Groups: True / False

Lab 1: Layout Advanced Editing Functions

30 minutes

- **Goals:**
  - Build a mask layout of a differential amplifier by using:
    - pcells
    - Instance creation
    - Manual placement
    - Routing functions



Create amplifier layout

Add and edit instance parameters

Abut instances to create interconnections

Align instances

Create interconnects

Import MPPs

Create Pins, Labels and cell boundary