

# SKILL CAD

---

*Accelerating Custom IC Layout*

**Flat Panel Layout Editor**

[WWW.SKILLCAD.COM](http://WWW.SKILLCAD.COM)

# Content

<b>User Quick Reference .....</b>	I
<b>About Skillcad Inc. ....</b>	1
<b>Supported Platforms.....</b>	2
<b>Installation.....</b>	3
<b>License Management .....</b>	6
FAQs on SKILLCAD License Management .....	8
<b>rRouter (Path) .....</b>	9
Examples.....	19
<b>rRouter(Polygon) .....</b>	32
Examples.....	38
<b>rChecker .....</b>	49
<b>iShape .....</b>	53
Example .....	54
<b>Align .....</b>	56
Align One By One .....	56
Align All Selected.....	60
<b>Mask Label .....</b>	63
<b>Merge Pins .....</b>	65
<b>Draw Line .....</b>	66
<b>Rule Extend .....</b>	67
<b>Round Corner.....</b>	68
<b>KITs.....</b>	70
Area Perimeter Calculator.....	71
Fix Offgrid .....	73
Filp within BBox.....	74
Full Selection .....	74
nCopy .....	74
Align/Distribute Pins(objs) .....	74

# SKILLCAD Flat Panel Layout Editor

## User Quick Reference

	<b>Icon Switcher</b> Click to toggle between SKILLCAD icon bar and CADENCE icon bar (in IC5)
	<b>rRouter (Path)</b> 1) Select Pins 2) Set Parameters. [ 3) Enter iGuide Points if using iGuide] 4) Click Preview 5) Adjust Parameters and Tune Routing Points 6) Apply
	<b>rRouter (Polygon)</b> 1) Select Pins and set Pin Extensions ] 2) Set Route Mode and Parameters. 3) Enter(Click) Routing Points 4) Adjust Parameters and Routing Width/Points 5) Preview and check Results 6) Finish Route
	<b>rChecker</b> 1) Click Start New Measurement 2) Draw a line cross starting ends of target lines 3) Draw a line cross ending ends of target lines. 4) Review/analysis results
	<b>iShape</b> 1) Select a drawing shape 2) Click at layout window to initialize the shape 3) Adjust Parameters/Cols/Rows 4) Click Apply to Finish Editing the shape and ready to start drawing a new shape
	<b>Align</b> OneByOne [1) Click on the reference object] 2) Click on the object to be aligned 3) Adjust offset/Orientation AllSelected: Use Align Arrows, Move Arrows and set Space buttons to adjust the placement of selected objects
	<b>iDRC</b> Click Apply/OK to check the given Space/Width design rules.
	<b>Mask Label</b> 1) Input the Label (Label Array) 2) Click at layout window to place Label(s)
	<b>Merge Pin</b> Draw a box over pins to be merged
	<b>Draw Line</b> Click in the same way as drawing Path.
	<b>Rule Extend</b> 1) Select the shape to be extended 2) Specify correct Space rule 3) Draw a closed polygon from the selected shape with reference to the outlines of the neighboring shapes
	<b>Round Corner</b> 1) Select bus 2) Select corners of the selected bus
	<b>Kits</b> Other utilities: Measure Area/Perimeter, fix offgrid, nCopy....

## **About Skillcad Inc.**

SKILLCAD is dedicated to developing routing and other automation tools for custom layout design. We provide correct and optimized by construction layout solutions to help customers achieve shorter time-to-market and higher yield.

SKILLCAD Inc. is a member of Cadence Design System connections program for third party developers. SKILLCAD tools run as advanced plug-in modules in the Virtuoso platform of Cadence Design System.

SKILLCAD owns the copy right of the entire menu including pictures and diagrams.

## **Supported Platforms**

Supported platforms are:

- LINUX 64bit (RHEL 3,RHEL4, RHEL5)
- LINUX 32bit (RHEL 3,RHEL4, RHEL5) and
- SUN Solaris 64bit.

Recommended Hardware configuration:

2G+ memory, 2.4G+ CPU speed.

Supported Cadence Virtuoso Versions:

- Version 5.1 or above for CDBA database.
- Version 6.1 or above for OA database.

SKILLCAD will maintain the compatibility to the new Cadence release.

## **Installation**

The following procedures are for the installation of SKILLCAD module into Cadence Virtuoso platform

### **i) Download Software**

SKILLCAD software is free to download from

<http://www.skillcad.com>

The default file name is “skillcad\_version.tar”.

### **ii) Unpack SKILLCAD**

After download, unpack the “skillcad\_version.tar” file with the command:

`tar -xvf skillcad_version.tar`

You will get two files: “readme.txt” and “stepRouter.tar”.

“readme.txt” contains the instructions on SKILLCAD installation and license management.

### **iii) Define SKILLCAD\_PATH environment variable.**

Assuming you are going to install Skillcad under “/tools/skillcad”, you are required to define SKILLCAD\_PATH environment variable in .cshrc for all SKILLCAD users:

`setenv SKILLCAD_PATH /tools/skillcad/`

#### **iv) Unpack stepRouter.tar**

Open a new shell terminal. Make sure \$SKILLCAD\_PATH is defined and you have write permission on the directory. It is strongly recommended to install SKILLCAD tools in an empty directory.

```
mv stepRouter.tar $SKILLCAD_PATH  
cd $SKILLCAD_PATH  
tar -xvf stepRouter.tar
```

The installation is done, make sure there are four directories under \$SKILLCAD\_PATH:  
bin/ doc/ lic/ source/

#### **v) Autoload the SKILLCAD software**

To autoload the Skillcad module when you start Cadence Virtuoso platform, add the following line **at the end** of the .cdsinit file:

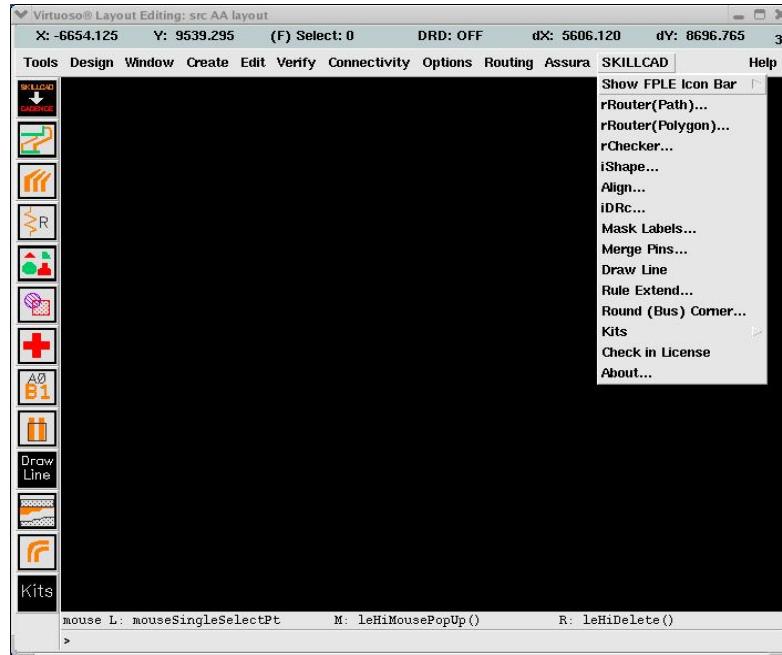
```
;*****loading SKILLCAD software *****  
when( getShellEnvVar("SKILLCAD_PATH") && isReadable(  
strcat( getShellEnvVar("SKILLCAD_PATH") "/source/skillcad.il"))  
;Pick one of the following SKILLCAD Packages for installation  
;SKILLCAD_PACKAGES = list( "IC Layout" ) ;; for IC Layout Automation Suite  
SKILLCAD_PACKAGES = list( "Flat Panel Layout" ) ;; for Flat Panel Layout Editor  
;SKILLCAD_PACKAGES = list( "All" ) ;; Default, all SKILLCAD modules  
load( strcat( getShellEnvVar("SKILLCAD_PATH") "/source/skillcad.il"))  
);when
```

#### **vi) license**

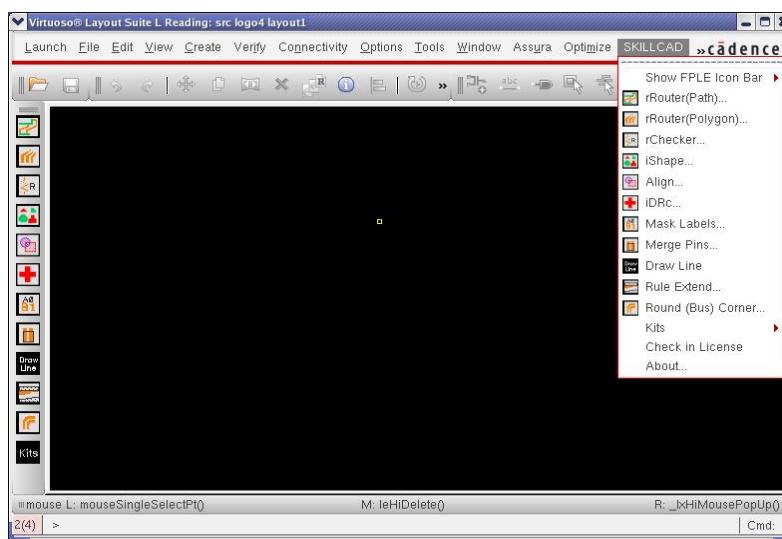
Follow the instructions in the given license file to activate the SKILLCAD licenses.  
SKILLCAD License is controled by Flexlm.

## vii) Test Installation

Now you can start Cadence Virtuoso platform. If the installation is correct, you should see the "SKILLCAD" pulldown menu in a layout window, as show in the following pages. If you did not see the "SKILLCAD" pulldown menu, please check the above installation procedures, and check the CDS.log file if there are error messages while loading your .cdsinit file.



SKILLCAD Flat Panel Tool Interface in IC5



SKILLCAD Flat Panel Tool Interface in IC6

## License Management

SKILLCAD uses FLEXnet (Version 11.4, a product of Microvision Inc.) licensing tool. FLEXnet is the de facto standard license management tool in the EDA industry. Please refer to the Microvision's FLEXnet licensing end user guide (you can download from [www.microvision.com](http://www.microvision.com) or [www.skillcad.com/doc/LicensingEndUserGuide.pdf](http://www.skillcad.com/doc/LicensingEndUserGuide.pdf)) to learn the details of Flexnet licensing management.

Here is an example of a typical SKILLCAD license file:

```
#SKILLCAD LICENSE FILE
#FLEXnet version 11.4
#
#----action 1:
#replace $ServerHostName$ with the real hostname of your license server
#replace $SKILLCAD_DEAMON_PATH$ with:
#For Linux platform:
#${skillcad installation path}/bin/linux/
#For sun platform:
#${skillcad installation path}/bin/sun/
#
#----action 2:
#Add this file to the beginning of LM_LICENSE_FILE environment variable(Recommended)
#or define a new environment variable SKILLCAD_LIC_FILE to this file
#
#----action 3:
#use the following lmgrd (flexlm version 11.4 or up) to start the license
# for linux platform
# $SKILLCAD_PATH/bin/linux/lmgrd
# for sun platform
# $SKILLCAD_PATH/bin/sun/lmgrd
#####
SERVER $ServerHostName$ 001c23afac78 5888
VENDOR SKILLCAD $SKILLCAD_DEAMON_PATH$/SKILLCAD
USE_SERVER
FEATURE stepRouter SKILLCAD 1.0 31-jul-2008 5 DUP_GROUP=UHD \
    ISSUED=17-jun-2007 SIGN="01C5 0256 AA24 D07B 0F4D FAEC B14F \
    3E9C 2830 B115 B600 5ACA AB81 8DB3 0052 5E0B F8FB 63A7 66ED \
    47FB 73EF"
```

You are recommended to follow the actions in the comments to change the license file first and start the daemon. If you know well about FLEXIm license management, you can also start the license daemon in your own way.

### **License Check-out/Check-in**

SKILLCAD license will be checked out at the first time when you use SKILLCAD tools to draw layout. Same user at the same machine and same display will use one (DUP) license. The license will be checked-in when you exist Virtuoso.

The license can also be manually checked in from the “SKILLCAD” pulldown menu, “**Check in License**”. Since DUP feature is enabled, you may need to do license check-in for all sessions you are running on that machine. The checked-in licenses will be available in the licenses pool in about 30 seconds (license linger time).

A valid Cadence Virtuoso Layout Editor license is required when you using SKILLCAD tools to draw layout, since Skillcad tool runs as an add-on module in Virtuoso platform. There is no affect or conflict between SKILLCAD commands and normal Cadence Layout Editor commands.

## **FAQs on SKILLCAD License Management**

**1) *Q: When will the license be checked out?***

A: SKILLCAD license will not be checked out until when the user starts using SKILLCAD tools. Once a license is checked out, the heartbeat interval is 10 Mins. The license status will also be checked irregularly while the program is running. The user may irregularly see “Checking out SKILLCAD license...” message in the CIW while using SKILLCAD commands.

**2) *Q: Will the license automatically checked in after certain idle time?***

A: Yes, it will be automatically checked if the idle time is more than 14 hours.  
The license can also be manually checked in at anytime from the “SKILLCAD” pulldown menu, “Check in License”

**3) *Q: If the user opens multiple Cadence sessions, will he/she checkout multiple SKILLCAD licenses?***

A: No. The user will only checkout one SKILLCAD license.  
The DUP policy of SKILLCAD license is UHD, that is, same user at same machine and same display port will only checkout one SKILLCAD license.

**4) *Q: What will happen if checking out license fails? (Such as server down, no more licenses)***

A: If the user attempts to use SKILLCAD functions during such event, the program will try to request and wait for the license server’s response for a maximum time about 2 Mins.

There is no effect of SKILLCAD license failure on Cadence (as long as Cadence license is OK) or other third party’s tools.

## rRouter (Path)

rRouter is designed for connections from driver pins to flat panel pins. Since it uses path in routing, the width of one route will be the same from start to end.

### ◆ Pins

The command offers two ways to get pins:

#### a) Get Pin By “Selection”

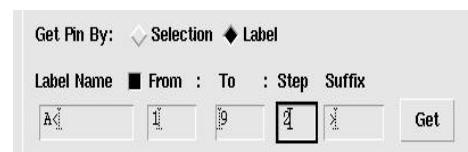
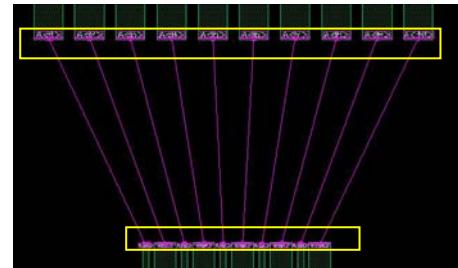
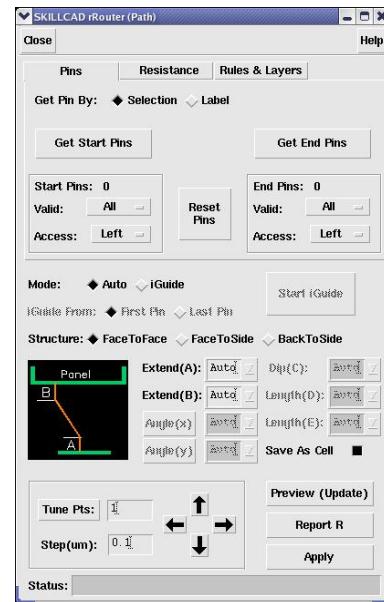
Click “Get Start Pins” or “Get End Pins” button,

Use mouse to draw a selection box over pins. Pins can be in different hierarchy levels. If multiple layers are found under the selection box, you will be prompted to select the pin layer. You can select a subset (**Odd**, **Even**, **1<sup>st</sup> Half**, **2<sup>nd</sup> half**) of pins as “Valid” pins to be connected. The number of valid “Start Pins” / “End Pins” has to be matched for routing.

“Access” direction determines from which side the pins can be connected. It also determines the order of pins and the routing “Structure” in “Auto” mode. Start Pins are indexed (1 2 3) from the right most pin (assuming you face to the access direction of start pins). End pins are also indexed (1 2 3) from the left most pin (assuming you face to the access direction of end pins). A flight line will be drawn between each matched pair of start/end pins.

#### b) Get Pin By “Label”

Input the label names and click “Get” button. The program will parse the input to get the individual pin name, and search labels at the top level. Floating labels

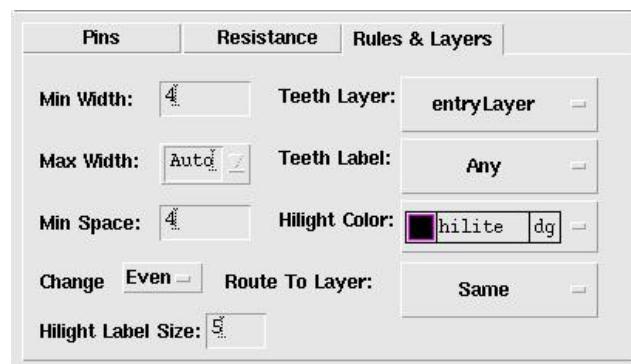


(with no pin layer underneath) will be ignored. You can set a specific label layer and teeth layer (in “**Rules & Layers**” tab), so that program will only search labels on the given label layer as valid teeth labels. Each pin name should have exactly two labels, one for the start pin and the other for corresponding end pin. The command will separate the matched labels into two groups: one as start pin labels, and the other as end pin labels. Then pins will be indexed according to their access direction. A flight line will be drawn for matched pins

**Note:** Pins are matched by order, not by the label names, since the program cannot handle the case when start pins and end pins are not in same order.

#### ◆ Rules & Layers

Make sure the given “**Min Width**” and “**Min Space**” meet the design rule and are even multiples of the layout grid.



**Note:** The command will use:

“**Min Space**” + 2\*grid

as the space between non-orthogonal segments.

The program will internally calculate an Allowed\_Max\_Width based on the “**Min Space**” and width and space of the start/end pins. The smaller one of “**Max Width**” (If not “**Auto**”) and Allowed\_Max\_Width will be the actual max width in routing.

“**Teeth Layer**”: Layer name for start/end pins, and the command will use this layer for routing. “**entryLayer**” means the current LSW layer.

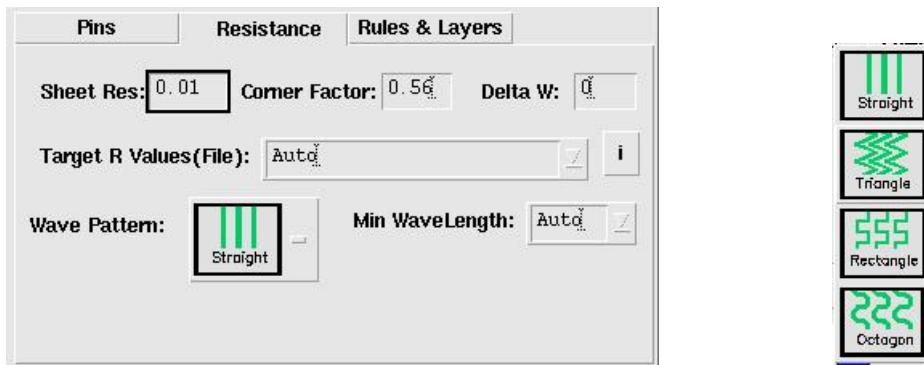
“**Teeth Label**”: The layer is used for pin labels. It will be used when get pins by label.

“**Hilight Color**”: The color of the layer will be used for flight lines and highlight labels.

“Hilight Label Size”: set the label size for pin index or resistance value.

“Change Even/Odd connection to Layer:” If set to “Same”, both even/odd pins will use “Teeth Layer” for routing. Otherwise, even or odd connections will be drawn in the given layer.

## ◆ Resistance

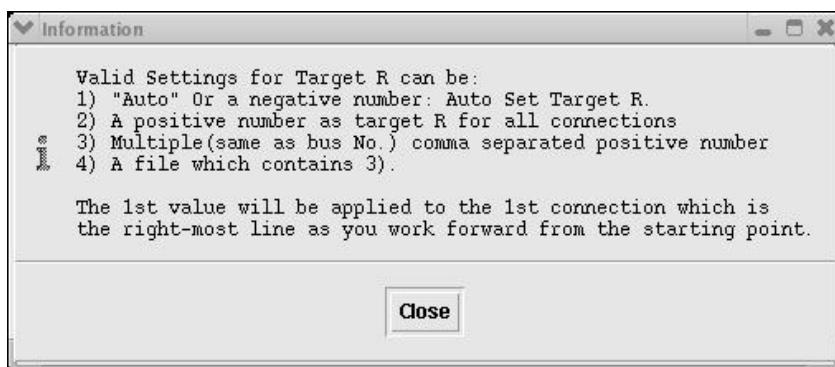


The program will use the following equation to calculate the resistance of a path:

$$\text{Path\_R} = \text{"Sheet Res"} * \text{pathLength} / (\text{pathWidth} - \text{"Delta W"}) - (1 - \text{"Corner Factor"}) * \text{pathTurns}$$

To ignore the “Corner Factor”, set it to 1.0.

Click the button “i” will tell you the ways to set “Target R”:



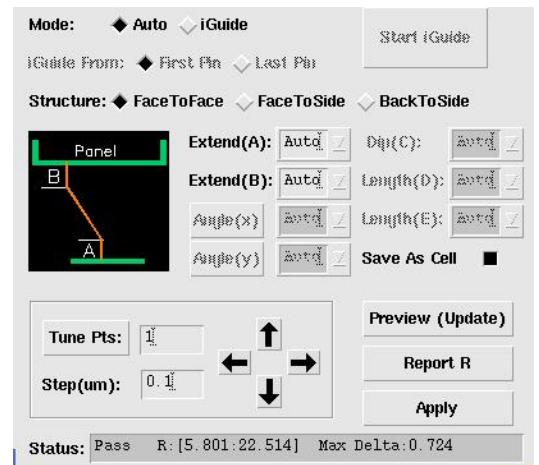
The program will first try to adjust width to meet target R. When the width reaches minimum width, it will introduce “Wiggle Pattern” (if it is not “Straight”) to increase the total routing length and thus increase the resistance. You can increase “Min WaveLength” to reduce the number of wiggles (thus reduce the total points/vertex in one connection).

Path rRouter provides two routing modes to do routing:  
**“Auto”** and **“iGuide”**

### i) Auto

“Auto” mode is for predefined common structures.

Routes for these common structures have fixed number of points and controlled by pre-defined parameters. The program will automatically pick the right structure after you select pins and set pin access directions.



#### ◆ FaceToFace

Two parameters are predefined for “**FaceToFace**” routing as illustrated in the form.

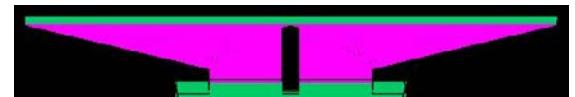


“**Extend(A)**” is the minimum pin extension for starting pins.

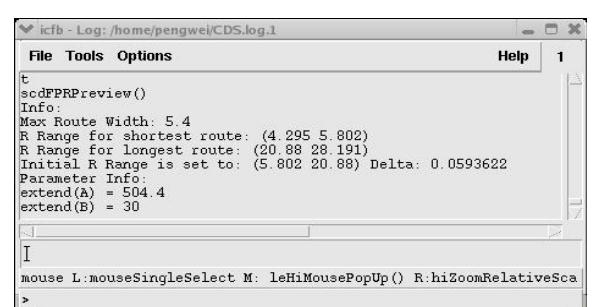
“**Extend(B)**” is the fixed pin extension for end pins (all end pins have same extension)

You can input a value for “**Extend(A)**” /**Extend(B)**” or use “**Auto**”. The “**Auto**” value for “**Extend(B)**” is half of the end pin pitch. The “**Auto**” value of “**Extend(A)**” is determined by the vertical distance from start pins to end pins and other routing parameters like “**Min Width**”, “**Min Space**”, “**Max Width**” and “**Target R**” etc.

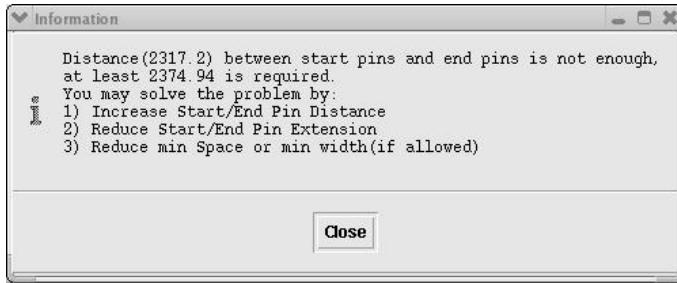
Click “**Preview(Update)**” button to preview the routing result. The preview result is drawn as window highlight, not as shapes in the cellview database.



You can see the resistance information and routing parameters in the CIW window. Meanwhile, the routing “**Status**” is shown at the bottom of the command form.



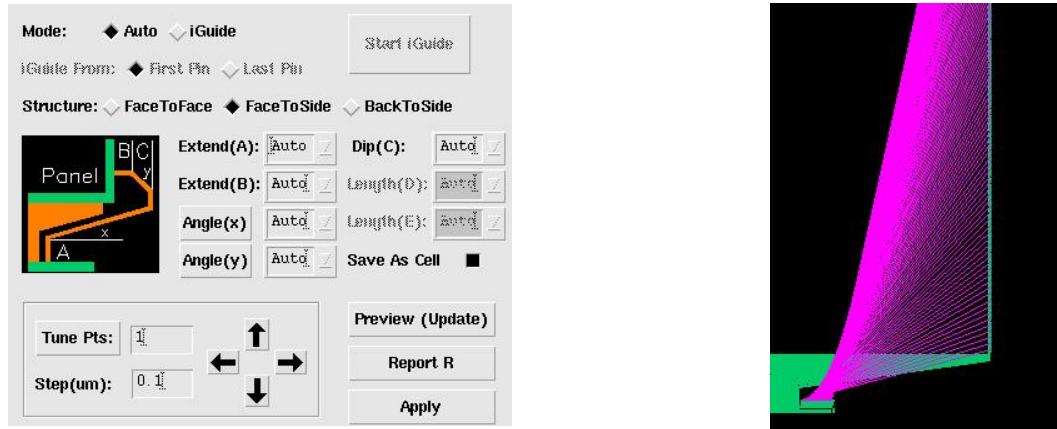
In case the vertical distance between start pins and end pins is not enough for routing. The program will stop routing and give you a warning message and instructions like:



**“Tune Pts”** is not applicable for **“FaceToFace”** auto routing. You can tune routing results by adjusting **“Extend(A)”**, **“Extend(B)”** and routing Width/Space etc.

#### ◆ FaceToSide

Five parameters are predefined for **“FaceToSide”** routing as illustrated in the form.



“Extend(A)” is the Maximum pin extension for starting pins.

“Extend(B)” is the fixed pin extension for end pins (all end pins have same extension)

“Dip(C)” is to set the vertical distance from the extended end pin to third routing segment which is parallel to the access direction of the start pins.

“Angle(x)” determines the first turning angle from the extended start pins.

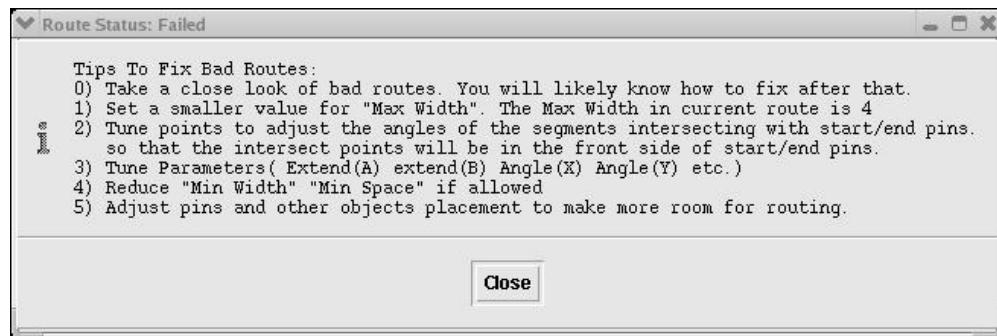
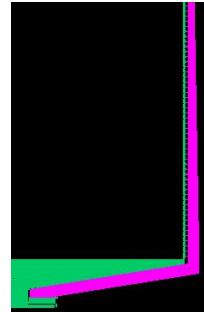
“Angle(y)” determines the turning angle from the extended end pins.

You can either input a value for “**Extend(A)**” /**Extend(B)**” or use “**Auto**”. The “**Auto**” value for “**Extend(B)**” is half of the end pin pitch. The “**Auto**” value of “**Extend(A)**” is determined by the distance from the innermost start pin to the closest existing routing(if any) and the maximum allowed “**Angle(x)**” (Mainly determined by the position of lowest end pin). The program will automatically get maximum “**Extend(A)**” and accurate “**Angle(x)**” parallel to the existing FaceToFace connections.

You can also click “**Angle(x)**” button, then click on an existing routing segment to get correct “**Angle(x)**” and make the routing parallel to that existing segment. So as to “**Angle(y)**”. Typically, “**Angle(x)**” is around 5-8 degrees, and “**Angle(y)**” is around 0-3 degrees. Set “**Angle(y)**” = 0 will make sidewall routes orthogonal.

The program cannot guarantee that the “**Auto**” values are the best for tour routing. The actual values of these parameters will be printed in the CIW after you click “**Preview(Update)**” button.

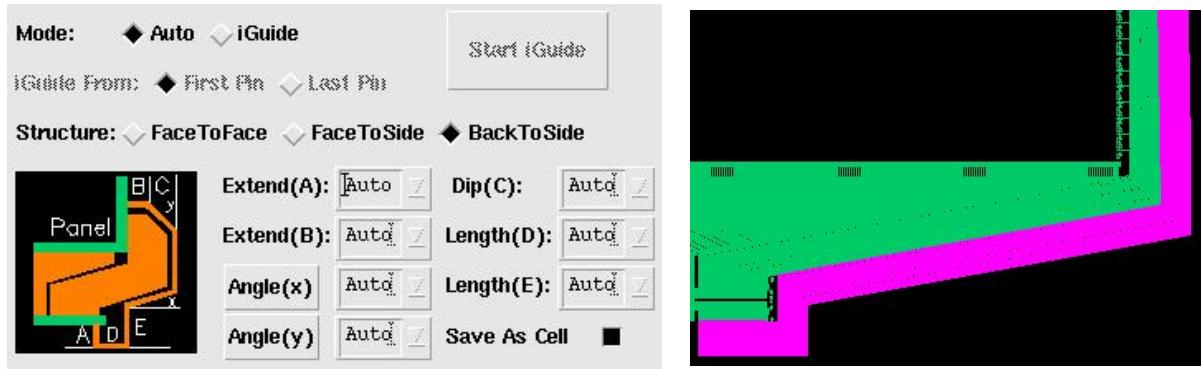
Click “**Preview(Update)**” button to preview the routing result. Again, the preview result is drawn as window highlight, not as shapes in the cellview database. In case the cannot finish routing correctly, the bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins) will be highlighted in (“marker” “warning”) layer, and the following message will pop up telling you how to fix bad routes.



“**Tune Pts**”: please refer to tune points in “**iGuide**” mode.

### ◆ BackToSide

Seven parameters are predefined for “**BackToSide**” routing as illustrated in the form.



“**Extend(A)**” is the minimum pin extension for starting pins.

“**Extend(B)**” is the fixed pin extension for end pins (all end pins have same extension)

“**Dip(C)**” is to set the vertical distance from the extended end pin to fifth routing segment which is parallel to start pins.

“**Angle(x)**” determines the first turn angle of the 4<sup>th</sup> segment.

“**Angle(y)**” determines the turn angle from the extended end pins.

“**Length(D)**” defines the length of the 2<sup>nd</sup> routing segment (parallel to end pins)

“**Length(E)**” defines the length of the 3<sup>rd</sup> routing segment (parallel to start pins)

The way of obtaining “auto” value of “**Extend(B)**” “**Length(E)**” “**Angle(x)**” “**Angle(y)**”

“**Dip(C)**” is similar as obtaining “**Extend(B)**” “**Extend(A)**” “**Angle(x)**” “**Angle(y)**”

“**Dip(C)**” in “**FaceToSide**” structure. The “Auto” value of “**Extend(A)**” is about half of start pin pitch. The “Auto” value of “**Extend(D)**” is about 20 times of start pin pitch.

Please refer to “**FaceToSide**” structure for other steps.

## ii) iGuide

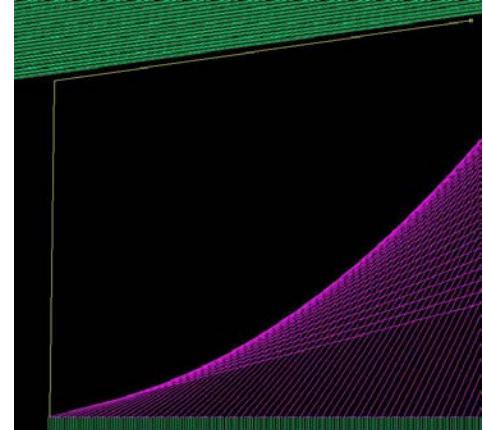
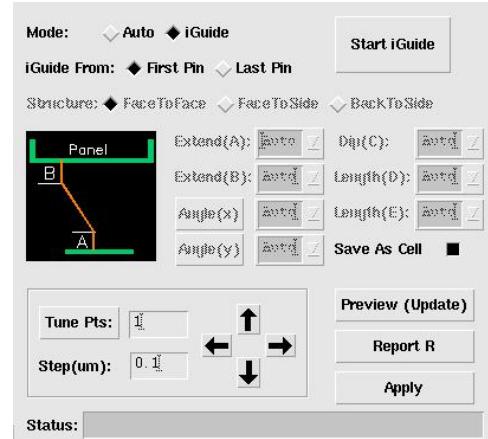
iGuide Routing mode is good for any structures. There is no limit on routing points and angles. The routing results can easily be tuned interactively.

The main idea of iGuide is: After you interactively input (click) the routing points for the connection of “First Pin” or “Last Pin”, the route will use this as the guide and intelligently generate proper connections for all pins. The guiding points can be tuned on fly during “Preview”.

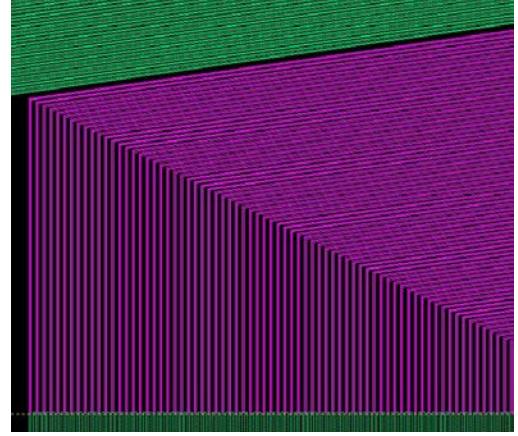
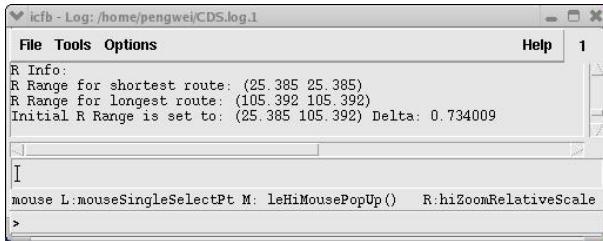
First selecting pins as in “Auto” mode. Normally, you should start iGuide from the side already occupied by other objects and you want to keep your routing as close as possible to them.

The first point of iGuide will be always snapped to the center of “First Pin” or “Last Pin” of start pins (up to the selection of “iGuide From”). You can continue the route from the first point as if you were drawing a path. Since the program cannot determine the exactly width of the guided connection at that time, you’d better keep enough margin of your guide line from existing objects, although you can tune them during preview stage. Keep entering the guiding points and hit key “Enter” or “Double Click” when you reach the corresponding end pin ( NOT required to click at the center of end pin) to finish iGuide.

The first and last segment of the guiding line will be replaced by start/end pin extension line, so it is not required to make the first/last segment parallel to pin access directions.

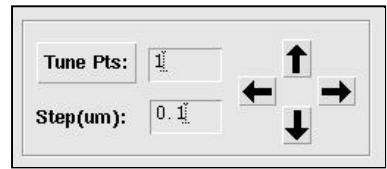


The routing results will be showed for your preview immediately after finishing iGuide. And the actual R info is shown in LSW (when target R is -1) and “Status” field in the command form.

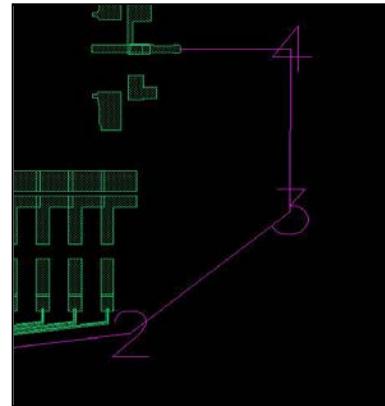


In case there are bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins) which are highlighted in (“marker” “warning”) layer, you can tune the guide points to fix them.

Click the “**Tune Pts**” button. The points of guiding line will be indexed by 1,2,3... (You can change the “**Hilight Label Size**” in the “**Rules and Layers**” tab to see the index comfortably),



Type the index of the point or multiple points you want to tune, and input a proper value for “Step” based on your measurement or estimation.



Each Click of “**Up**”/ “**Down**”/ “**Left**”/ “**Right**” button will move the specified points to the corresponding direction by a “Step”.

The routing result will be updated immediately after being tuned.

iGuide will automatically flip the routing direction of the last segment if the relative pin position of the starting pin and ending pin is changed.

#### ♦ Report R

Click “**Report R**” will report the resistance for each route in the rChecker form. Please Refer to command “rChecker” for details.

#### ♦ Apply

Click “**Apply**” to draw the final results to the cellview database. You will be prompted to input a cell name (or a group name in IC6) when “**Save As Cell**” (“**Save As Group**” in IC6) option is on.

If there are bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins), they will be drawn in (“marker” “warning”) layer.

After “**Apply**”, the drawn wires cannot be tuned through the rRouter form, and the Start/End pins information is reset for you to start a new route.

## Examples

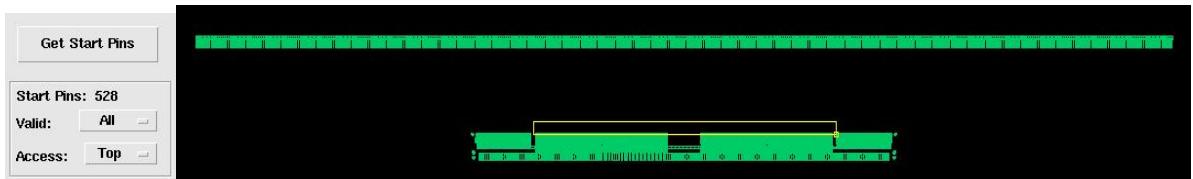
### 1) Auto Mode, FaceToFace Structure

Start rRouter(path) command.

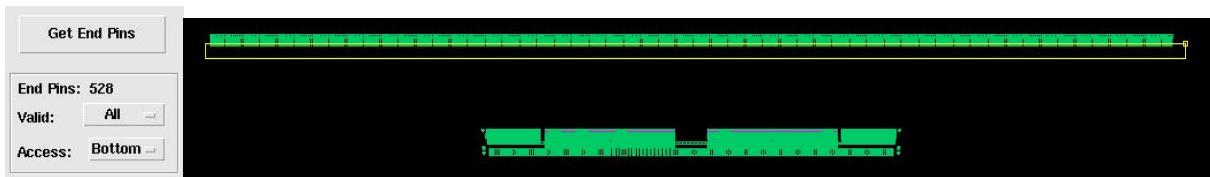
Set “**Target R Values**”: “**Auto**” and proper “**Min Width**”, “**Min Space**” etc rules.

Set Route “**Mode**”: “**Auto**”. Set “**Wave Pattern**” to “**Straight**”

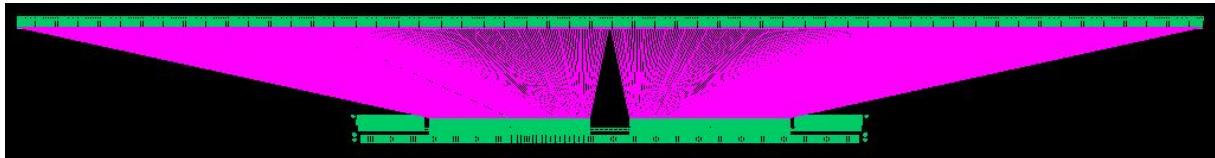
Click “**Get Start Pins**”, and draw a selection box over start pins ( Driver Pins). Start pins will be indexed, and the total number of start pins will be shown in the command form. Make sure the pin “**Access**” direction is “**Top**” if the program does not automatically set it correctly.



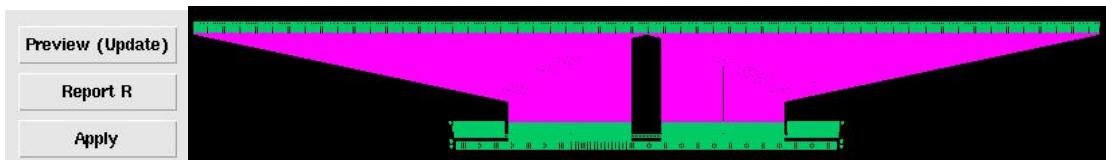
Click “**Get End Pins**”, and draw a selection box over end pins ( Flat Panel Pins). . Make sure the pin “**Access**” direction is “**Bottom**” if the program does not automatically set it correctly.



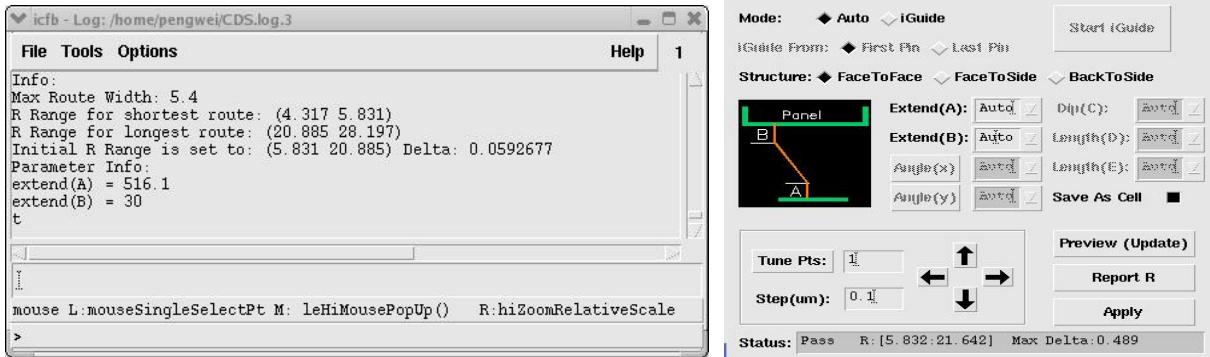
A faint highlight line is drawn between each matched pair of start pin/end pin. The program should automatically detect and set the “**Structure**” to “**FaceToFace**”



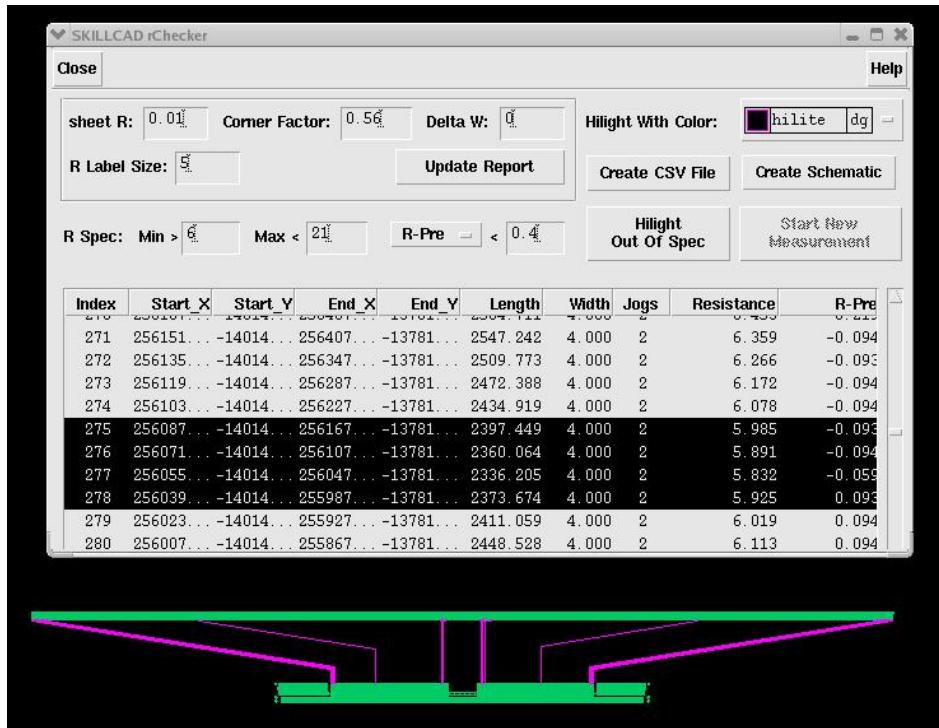
Click “**Preview (Update)**” button to preview the result.



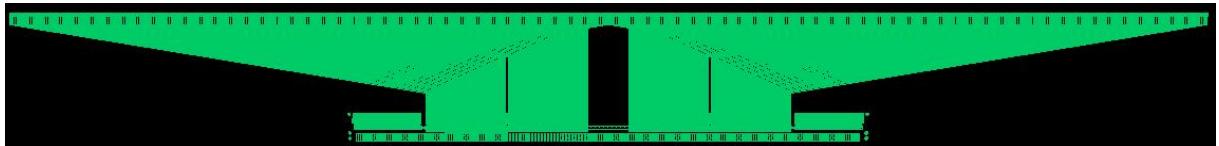
Initial R information (when Target R set to “Auto”), max routing width, and values of auto routing parameters are printed in LSW after “Preview”. And the “Status” in the command form is updated, telling you the routing status, the actual R range and maximum resistance difference of all adjacent routes.



Click “**Report R**”. The detail resistance information will be shown in SKILLCAD rChecker.  
(refer to rChecker for details)

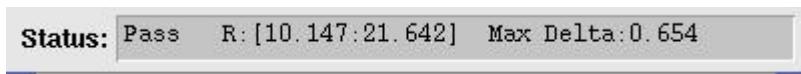
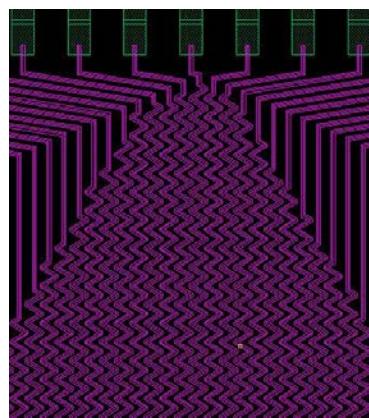
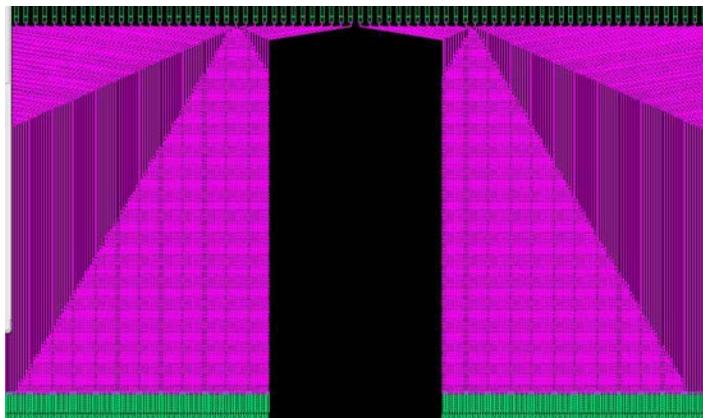


Click “**Apply**” to complete the route. If option “**Save As Cell**”/“**Save As Group**” in IC6 is on, you will be asked to enter a cell name/group name for the results.



### Triangle Wave Pattern:

Let's redo the previous example, and set wiggle pattern to “Triangle”. All other settings and steps are the same. You will see triangle wiggle patterns are created for short routes to increase their resistance.



Compare to the R range [5.832 21.642] in the previous example (“Straight” Wave Pattern), the smallest R becomes bigger, and the R range becomes narrower after using Triangle wiggle pattern.

“Triangle” wave pattern will only created for routes with minimum width. Since “Triangle” wave patterns of adjacent routes are stacked, to prevent DRC violation, the width of adjacent routes should be same (minimum).

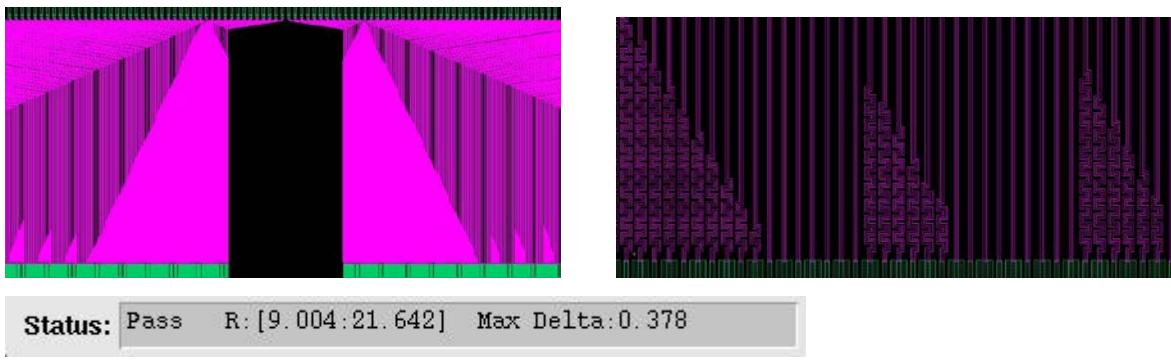
### Rectangle Wave Pattern:

Set wiggle pattern to “Rectangle”. All other settings and steps are the same. You will see rectangle wiggle patterns are created for short routes to increase their resistance.

Wave Pattern:



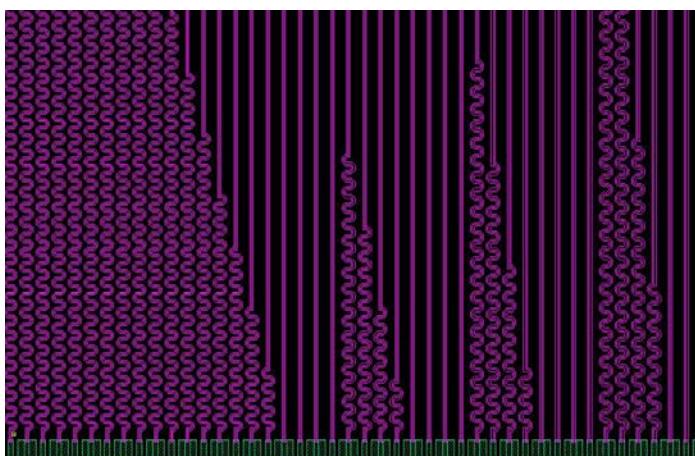
Rectangle will also be created when the routing width changes, in order to reduce the resistance difference between adjacent routes. Since route width has to be even multiple of grid, the resistance difference of two adjacent routes due to width change can be ~5% (Fox example, when min width is 4um and grid is 0.1um).



### Octagon Wave Pattern:

Similar to Wave Pattern, except octagon wave pattern has more turns and requires more space (pitch) between two adjacent routes.

Wave Pattern:



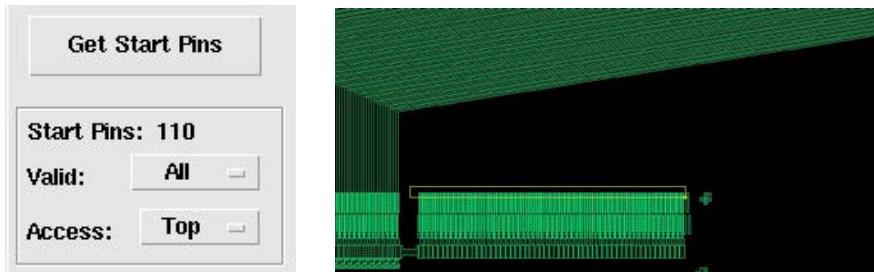
## 2) Auto Mode, FaceToSide Structure

Start rRouter(path) command.

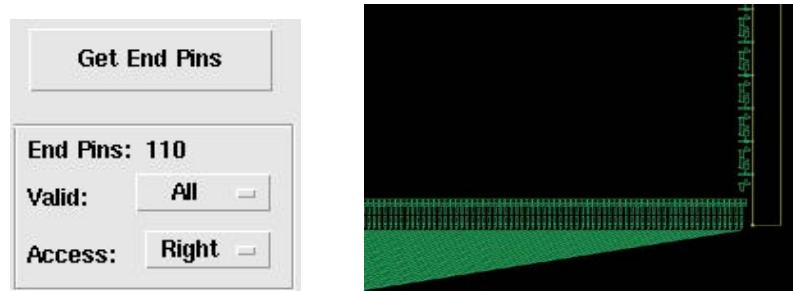
Set “**Target R Values**”: “**Auto**” and proper “**Min Width**”, “**Min Space**” etc rules.

Set Route “**Mode**”: “**Auto**”. Set “**Wave Pattern**” to “**Straight**”

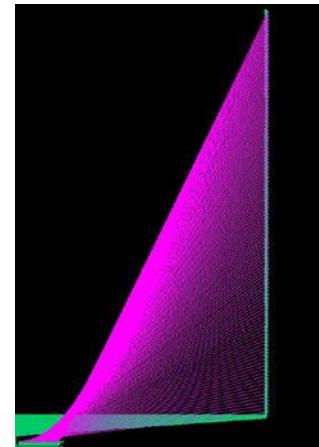
Click “**Get Start Pins**”, and draw a selection box over start pins ( Driver Pins). Start pins will be indexed, and the total number of start pins will be shown in the command form. Make sure the pin “**Access**” direction is “**Top**” if the program does not automatically set it correctly.



Click “**Get End Pins**”, and draw a selection box over end pins ( Flat Panel Pins). Make sure the pin “**Access**” direction is “**Right**” if the program does not automatically set it correctly.



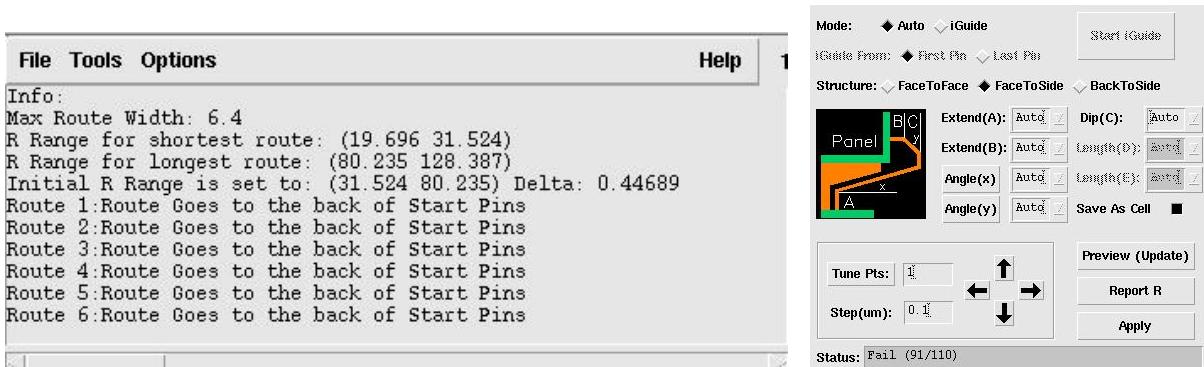
A faint highlight line is drawn between each matched pair of start pin/end pin. The program should automatically detect and set the “**Structure**” to “**FaceToSide**”



Click “Preview (Update)” button to preview the result. Bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins) are drawn in (“marker” “warning”) layer. And a pop message will give you instructions to fix them.

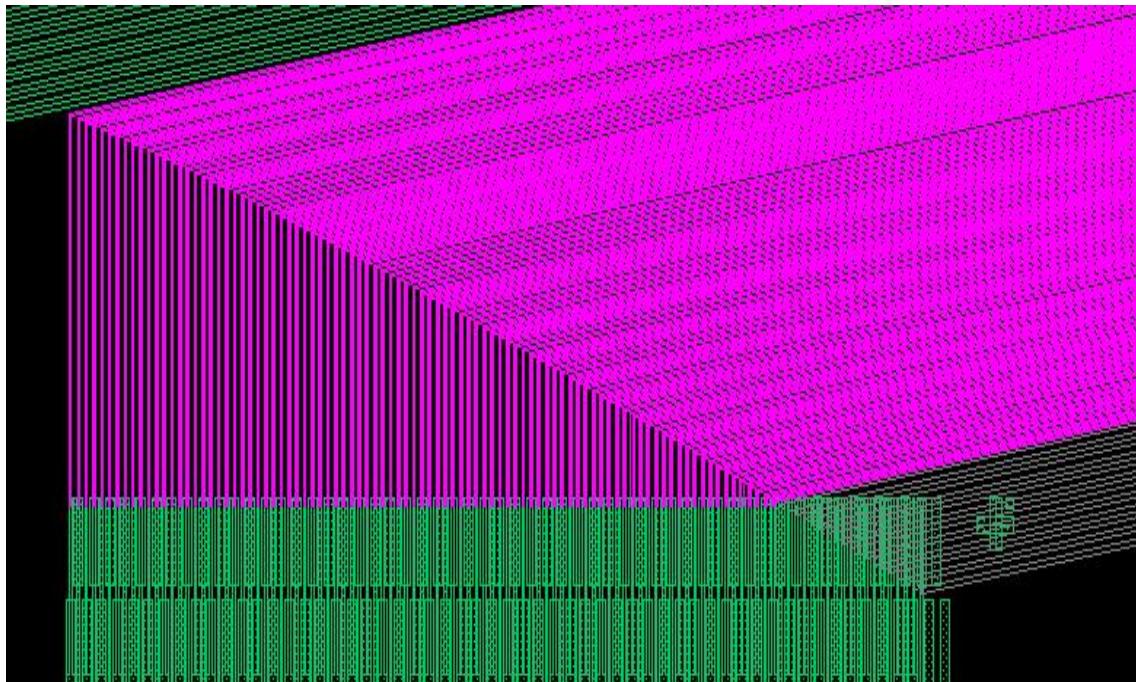


Initial R (when Target R = “Auto”) and other routing information are printed in CIW. Routing status is updated in the command form, telling you the routing is failed with 91 good routes out of total 110.



After zoom in, you will find that the bad routes in this example go to the backside of start pins (As told in the CIW), because the routing room at that region is limited by the pre-routed wires.

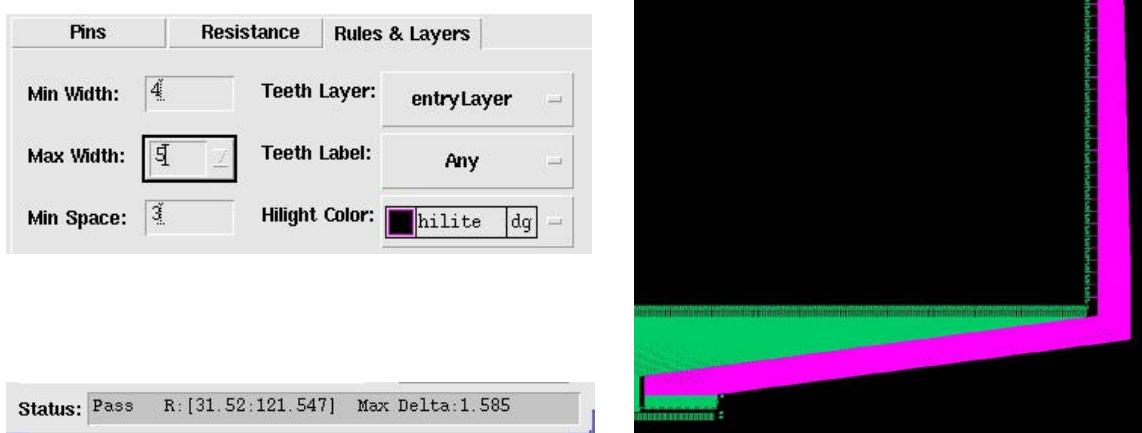
Meanwhile, you will see that the command can intelligently detect pre-routed lines and set correct “**Extend(A)**” (start pin extension) and “**Angle(x)**” so that the wires are close to and parallel to the existing lines.



The maximum width in this failed route is 6.4um as told in CIW and pop up message. To fix the bad routes (to reduce the total route range), you can first try out setting “**Max Width**” to the same value as “**Min Width**” ( That is, all wires will use “**Min Width**”), and click “**Preview**” to see whether there are no failed routes. If there are still some failed routes, the pin placement or the pre-routes wires may have to be adjusted to make more room. If there are no more failed routes, you can gradually increase the maximum to get a better result.

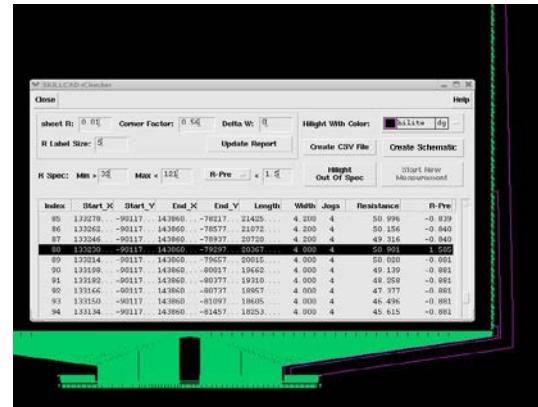
**Note:** Resistance range will decrease but total routing range will increase when “**Max Width**” increases. Or, total routing range will decrease but Resistance range will increase when “**Max Width**” decreases.

Bad routes in this example are fixed after setting “Max Width” = 5um (change from “Auto”)

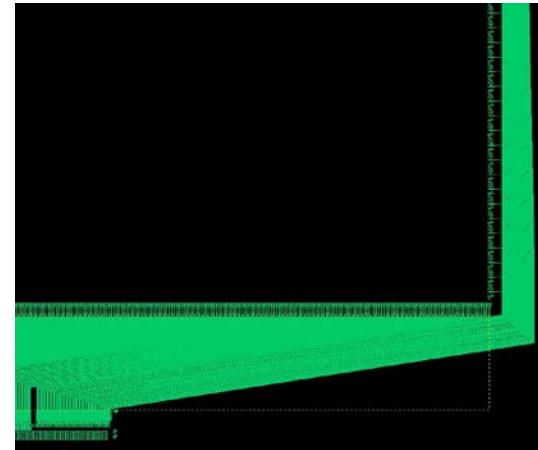


Click “Report R”. The detail resistance information will be shown in SKILLCAD rChecker.

(refer to rChecker for details)



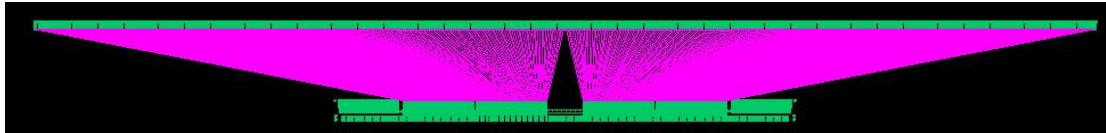
Click “Apply” to complete the route. If option “Save As Cell”/(“Save As Group” in IC6) is on, you will be asked to enter a cell name/group name for the results.



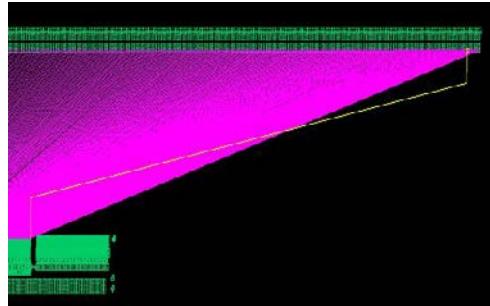
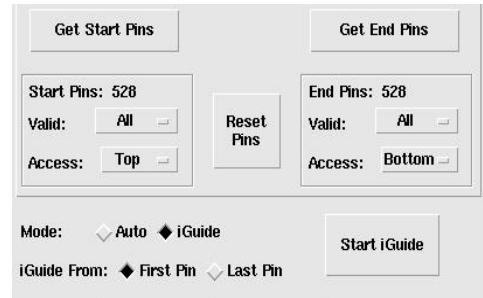
### 3) iGuide, FaceToFace Structure

Use the same case as in example 1.

Get Pins as in example 1.



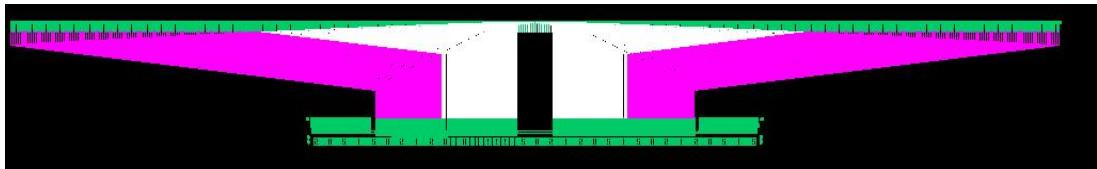
Select “Mode” to “**iGuide**”, and set “**iGuide From**” to “**First Pin**”. Move mouse in the layout window, you will see a rubber line starting from the center of first start pin. Click to continue the guided route from the first point as if you were drawing a path. Since the program cannot determine the exactly width of the guided connection at this time, you’d better keep enough margin of your guide line from existing objects, although you can tune them during preview stage. Keep entering the guiding points and hit key “**Enter**” or “**Double Click**” when you reach the corresponding end pin ( NOT required to click at the center of end pin) to finish iGuide.



The first and last segment of the guiding line will be replaced by start/end pin extension line, so it is not required to make the first/last segment parallel to pin access directions.

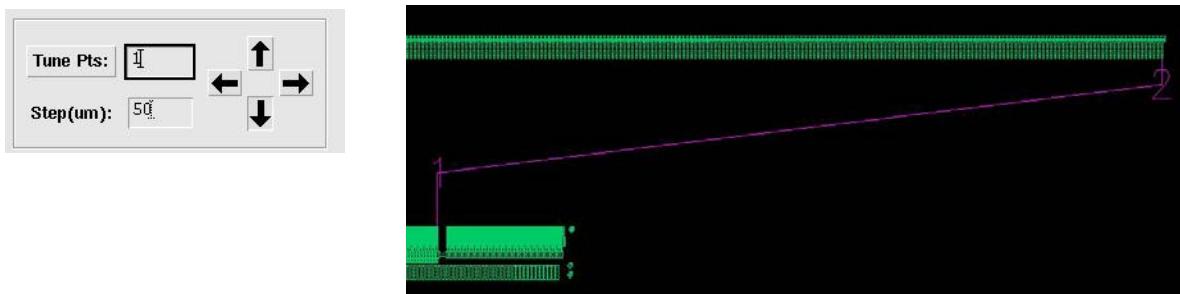
You can use “**BACKSPACE**” key to delete the previous entered routing point. If you cancelled the iGuide command (ESC key), you can start over by clicking “**Start iGuide**” button.

The program will show the entire routing result for your preview after you finishing iGuide. Bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins) are drawn in (“marker” “warning”) layer. And a pop message will give you instructions to fix them. Besides reduce the “**Max Width**”, the failed routes in iGuide can mostly be fixed by “**Tune Pts**”

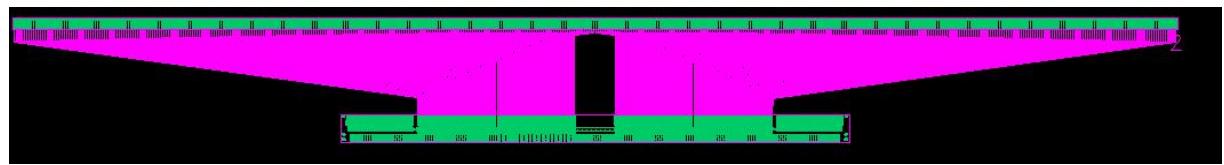


### Tune Points

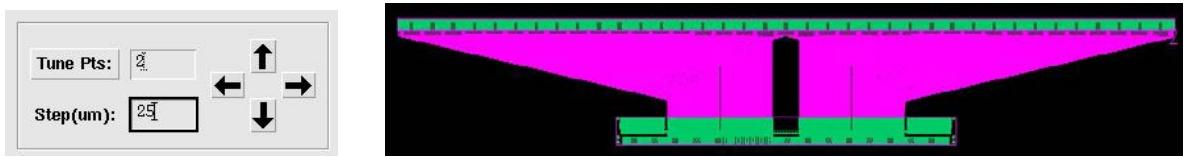
Click “**Tune Pts**” button, the iGuide points will be indexed (1 2 3...) orderly. The starting point (center of the start pin) and ending point (center of the ending pin) cannot be changed.



The bad connections in this example go to the backside of end pins. We can move point “1” lower to make more routing room. Measure how much the bad routes go beyond the end pins, you will get a good estimation for the tuning “**Step**”. Each Click of “**Up**”/ “**Down**”/ “**Left**”/ “**Right**” button will move the specified points to the corresponding direction by one “**Step**”.



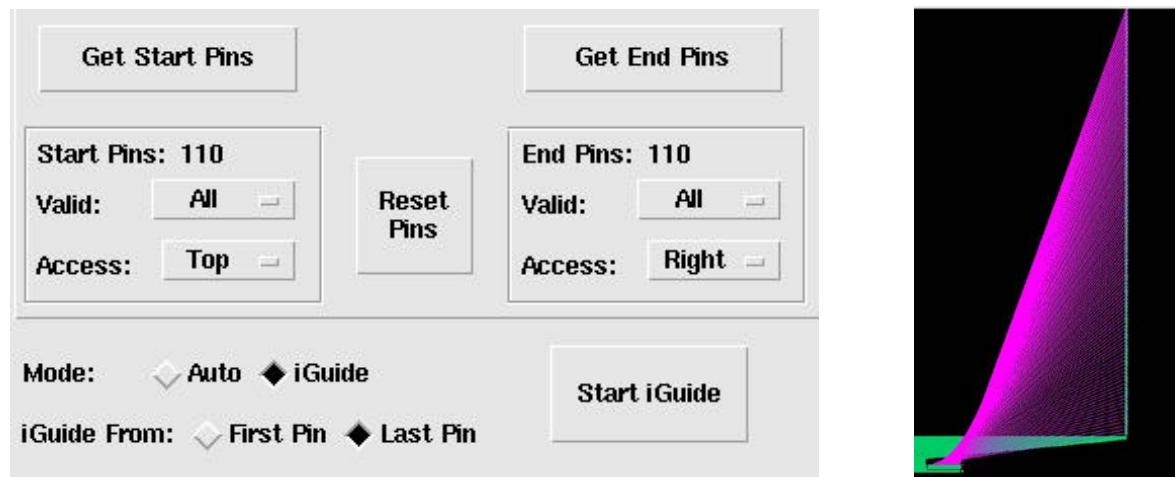
Reduce the “**Step**” to fine tune the result. Move the point “2” up a little bit make the result in this example better.



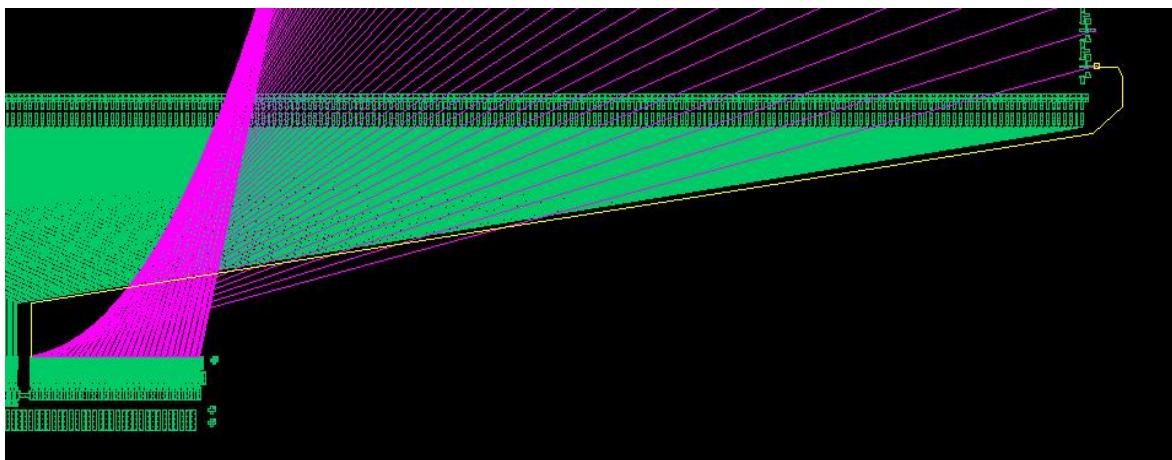
#### 4) iGuide, FaceToSide Structure

Use the same case as in example 2.

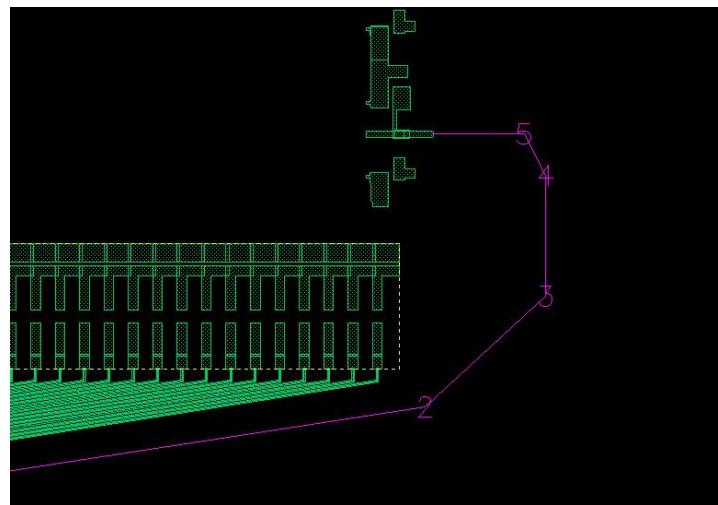
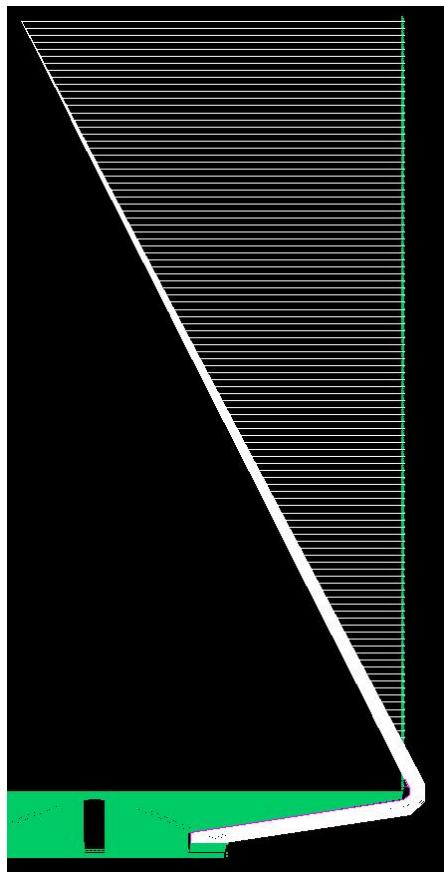
Get Pins as in example 2.



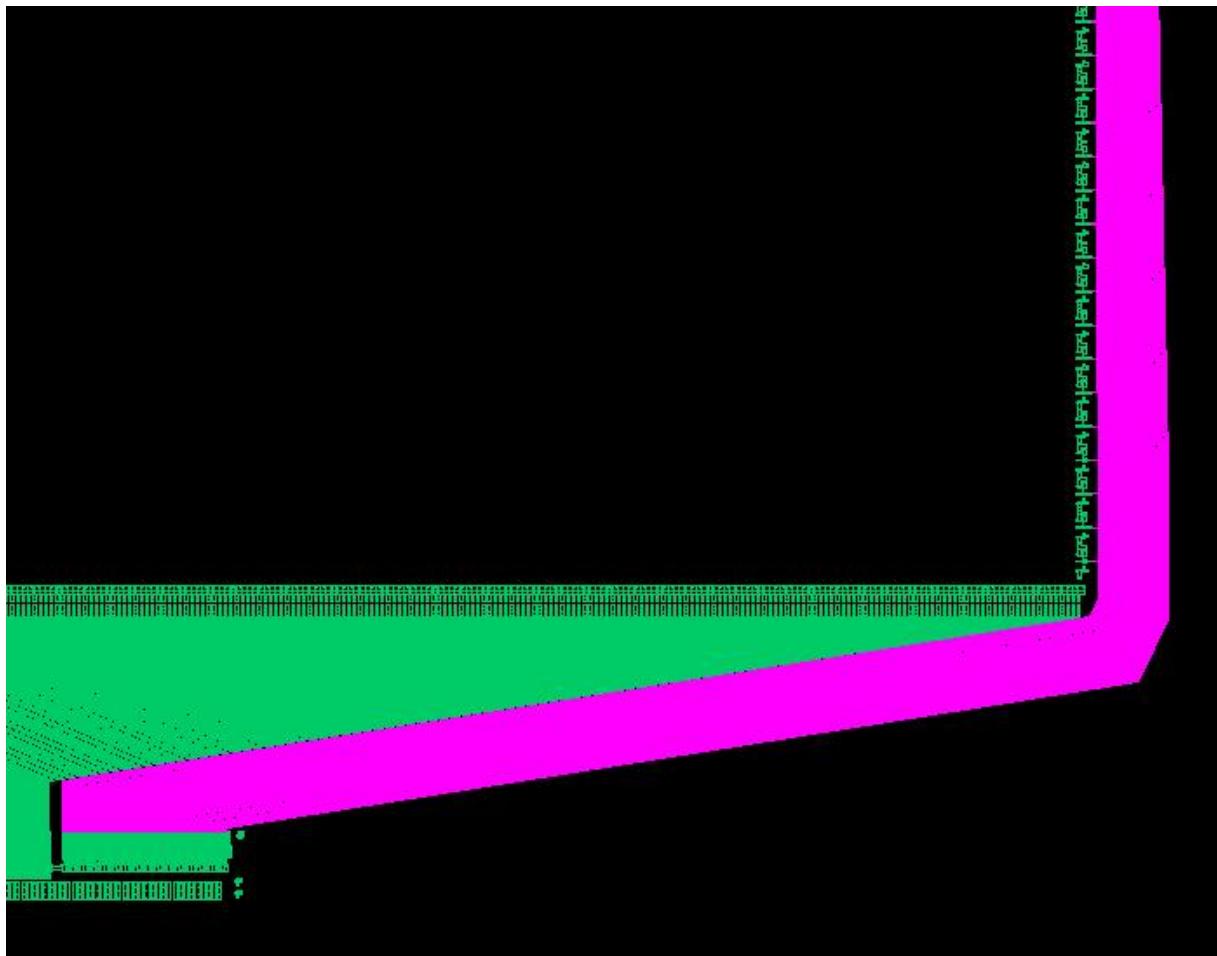
Refer to the steps in example 3, Start iGuide from “Last Pin”, and enter the iGuide points as shown by yellow rubber line.



The routing result will be shown for your preview after finishing iGuide. Bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins) are drawn in (“marker” “warning”) layer. There are many bad routes that connect to the backside of end pins. Click “**Tune Pts**” button, the iGuide points will be indexed (1 2 3...) orderly. The starting point (center of the start pin) and ending point (center of the ending pin) cannot be changed.



By looking at the bad routes, we need to tune the angle of segment between point “5” and point “4” --- specify “Tune Pts” = 5, “Step” = 10, and keep clicking “Right” Arrow to move point “5” rightward until all bad connections are fixed ( change to a small step for fine tuning). We can also move points 3,4,5 leftward together so that the sidewall routes will be closer to the end pins.



## rRouter(Polygon)

rRouter (Polygon) is an interactive polygon router. You can specify a routing range/width for each node, and create connections in four modes: “Equal R” “Ratio W” “Equal W” and “Extend”. rRouter is always prompt to enter routing points like “Create Path” command.

### ◆ Pins

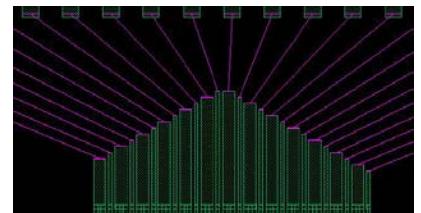
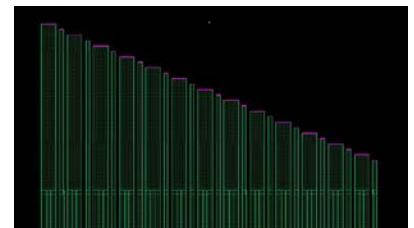
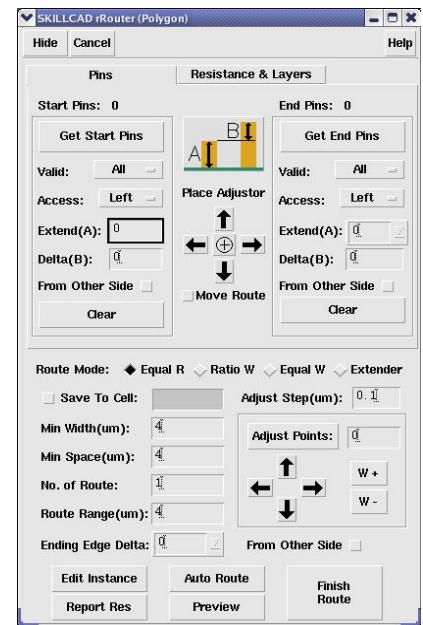
Click “Get Start Pins” or “Get End Pins” button,

Use mouse to draw a selection box over pins. Pins can be in different hierarchy levels. If multiple layers are found under the selection box, you will be prompted to select the pin layer. You can select a subgroup (**Odd, Even, 1<sup>st</sup> Half, 2<sup>nd</sup> half**) of pins as “Valid” pins to be connected. “Access” direction determines from which side the pins can be connected. Start Pins are ordered from the right most pin (assuming you face to the access direction of start pins). End pins are ordered from the left most pin (assuming you face to the access direction of end pins). A flight line will be drawn for each matched pair between start/end pins. “No of Route” and “Route Range” will be updated based on start pins info.

“Extend(A)” specifies the length of extension for the first pin ( or the last pin if “From Other Side” is on ).

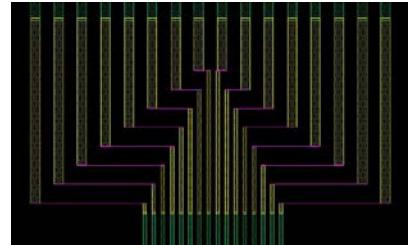
“Delta(B)” specifies the increment of extension for the next or previous pin depending on the option of “From Other Side”.

If there are face to face end pins, Pin Extension will automatically change the direction when the relative position of the corresponding start pin / end pin changes. Option



“From Other Side” is ignored when there are face to face end pins.

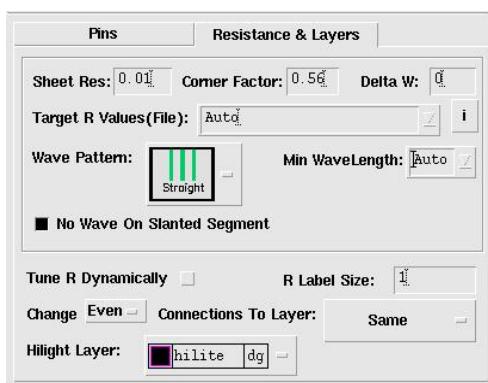
Set the end pin’s “Extend(A)” to “Align To Start Pins”. Option “From Other Side” is ignored in this case. Each end pin will be extended/Aligned to the same level of the corresponding start pin.



Polygon rRouter also includes a “Place Adjustor” to align/move start pins. Click an arrow button, the group of start pins (or the instance contains start pins) will be move by the given “Adjust Step” in the command form. The Center “+” button will align the center of start pins to the center of end pins. If you have created some routes from the start pins, these wires will also moved to keep connection with start pins when “Move Route” option is on.



#### ◆ Resistance & Layers



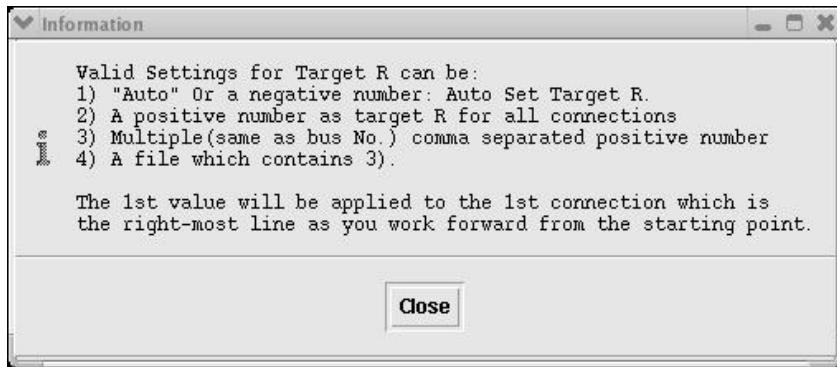
The program will use the following equation to calculate the resistance of a path:

$$\text{Path\_R} = \text{“Sheet Res”} * \text{pathLength} / (\text{pathWidth} - \text{“Delta W”}) - (1 - \text{“Corner Factor”}) * \text{pathTurns}$$

To ignore the “**Corner Factor**”, set it to 1.0.

In case of a polygon shape, the shape will be tiled into trapezoids and triangles first, the program will calculate resistance for each trapezoid and triangle to get the total resistance of the polygon shape.

Click the button “**I**” will tell you the ways to set “**Target R**”:



The program will first try to adjust width to meet target R. When the width reaches minimum width, it will introduce “**Wiggle Pattern**” (if it is not “**Straight**”) to increase the total routing length and thus increase the resistance. You can increase “**Min WaveLength**” to reduce the number of wiggles (thus reduce the total points/vertex in one connection).

Polygon rRouter is able to create wiggle patterns on Slanted segments when there are enough wiggling space and option “**No Wave On Slanted Segment**” is turned off.

“**Tune R Dynamically**”: in “Equal R” route mode, the program will take some time to tune the route result to meet R. To speed up, you can turn off this option in the middle of interactively entering routing points (The program will temporarily use “**Equal W**” mode), and turn on this option before you finish routing.

“**Hilight Layer**”: The color of the layer will be used for flight lines and highlight labels.

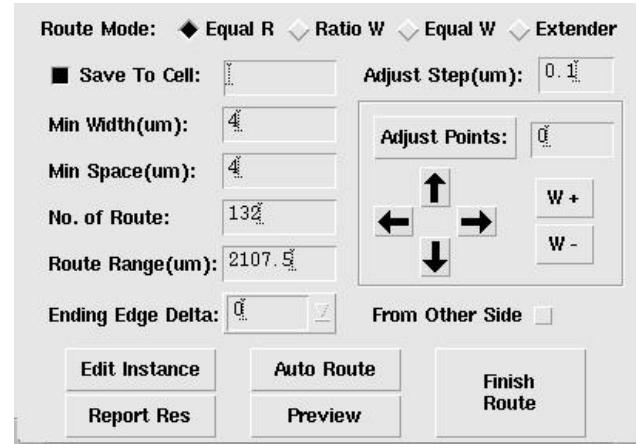
“**R Label Size**”: set the label size for resistance value.

“**Change Even/Odd connection to Layer:**” if set to “**Same**”, both even/odd pins will use “**Teeth Layer**” for routing. Otherwise, the corresponding connections will be the given layer for routing.

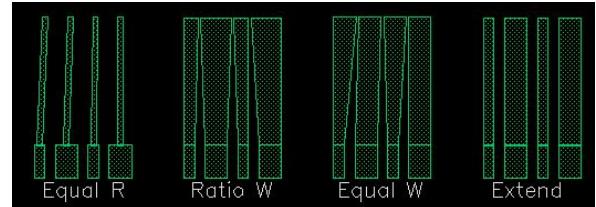
#### ◆ Route Mode

Polygon rRouter provides 4 routing Modes:

**“Equal R”:** The router will use the given R information in the “**Resistance & Layers**” tab, and try to tune each route by adjusting the width and/or introducing “**Wiggle Pattern**” to meet the given target R. When target R is “**Auto**” or a negative number, the program will first try to set a same target R for each route. If same R cannot meet, the command will set a constant R increment from the shortest route.



**“Ratio W”:** The width of each route will be pro-rated at each node according to the width of the corresponding start pin. “**Min Space**” will be maintained between neighbouring routes.



**“Equal W”:** The width of each route will be same at each node regardless of the width of the corresponding start pin. “**Min Space**” will be maintained between neighbouring routes.

**“Extender”:** Original Space between neighbouring start pins will be maintained in routing. The width of each route will be pro-rated at each node according to the width of the corresponding start pin. If the “**Route Range**” is kept the same as that at starting pins, the width of each route will be the same as the width of the corresponding pin.

“Save To Cell” will save the final routing result to the given **cell name** after you click “Finish Route”. The routing parameters and pin info etc will also be saved as cell properties.

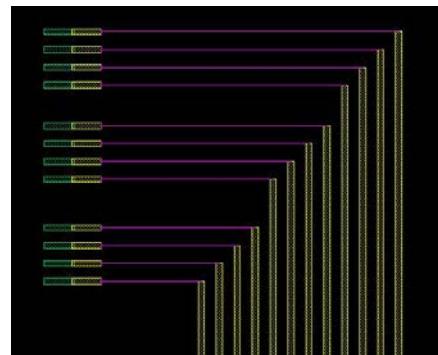
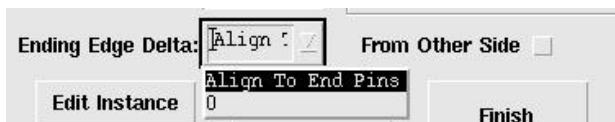
“Min Width” has to be an even multiple of your layout grid.

If the given “Route Range” is less than minimum route range which is calculated by:

“Min Width” \* “No. of Route” + “Min Space” \* (“No. of Route” - 1)

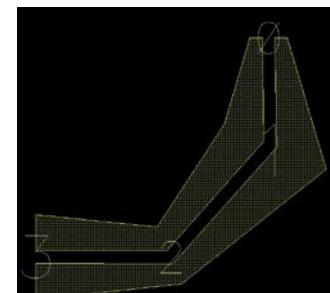
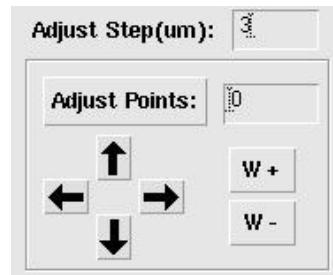
It will be set to minimum route range automatically.

“Ending Edge Delta” allows you to extend the ending of the route. If it is a number, each ending edge will be incrementally extended by this amount from the 1<sup>st</sup> or last route depending on the option of “From Other Side”. If it is set to “Align to End Pins”, the ending edge will be extended to the same level as the corresponding end pin.



The routing points and “Route Range” can be adjusted on fly.

Click the “Adjust Points” button. The points of routing coordinates will be indexed by 0,1,2,3... Type the index of the point or multiple points you want to tune. Input a proper value for “Adjust Step”. Each Click of “Up”/ “Down”/ “Left”/ “Right” button will move the specified points to the corresponding direction by one “Adjust Step”. Each click “W+”/ “W-” button to increase the width at specified node by the amount of one “Adjust Step”. The routing result will be update immediately after being tuned.



“Save To Cell” will save routing parameters and pin info etc as cell properties. Click “Edit Instance” button, and click the saved results, the program will reset command forms to the same conditions as the time you did the routing, and allow you to change the parameters, tune points/width to modified the results.

“Auto Route” is for predefined “FaceToFace” and “FaceToSide” structures as illustrated in rRouter(path) form. You have to set proper start/end pin extend/delta before using “Auto Route”. You may need to “Adjust points” or width after “Auto Route”. It is recommended to use “Auto” route in rRouter(path) which has better interface to control the result.

#### ◆ Report RES

Click “Report Res” will report the resistance for each route in the rChecker form. Please Refer to command “rChecker” for details.

#### ◆ Preview

Click “Preview” button, the program will connect the current routing edges to the end pins correspondingly and draw the result (not highlight) into the cellview. You can continue your routing, adjust points/width and/or delete routing points (BACKSPACE key) after “Preview”. If there are bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins), they will be drawn in (“marker” “warning”) layer.

#### ◆ Finish Route

Similar as “Preview”, except that the result is final (saved as cell if the option is on) and cannot be tuned after Clicking “Finish Route”.

You can also use “ESC” key after “Preview” to accept the result. ( Except that the result will not be save as cell even if the option is on)

## Examples

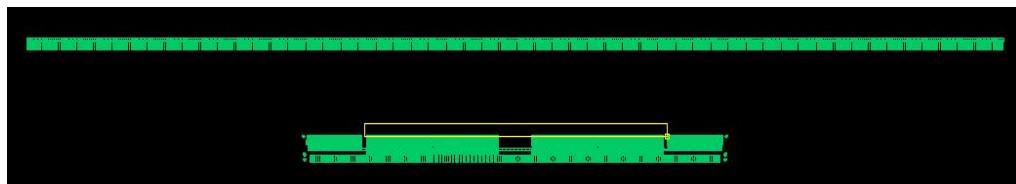
### 1) FaceToFace Structure

Start rRouter(polygon) command.

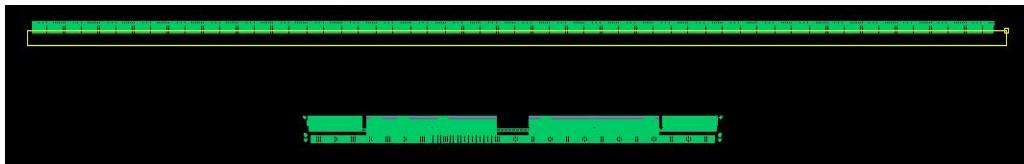
Set “**Target R Values**”: “**Auto**” and proper “**Min Width**”, “**Min Space**” etc rules.

Set “**Wave Pattern**” to “**Straight**”

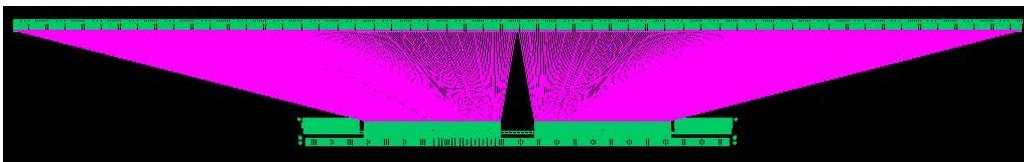
Click “**Get Start Pins**”, and draw a selection box over start pins (Driver Pins). The total number of start pins will be shown in the command form. Make sure the pin “**Access**” direction is “**Top**” if the program does not automatically set it correctly. The “**No. of Route**” and “**Route Range**” are updated to the number of start pins and total range of start pins respectively, and the first routing point is snapped to the center of start pins. You cannot use “**BACKSPACE**” key to cancel this first routing point. You need to “**Clear**” the start pin to start over.



Click “**Get End Pins**”, and draw a selection box over end pins ( Flat Panel Pins). . Make sure the pin “**Access**” direction is “**Bottom**” if the program does not automatically set it correctly.

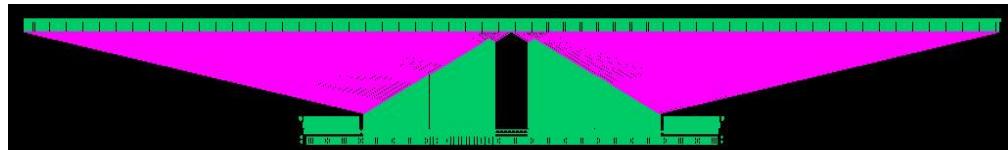
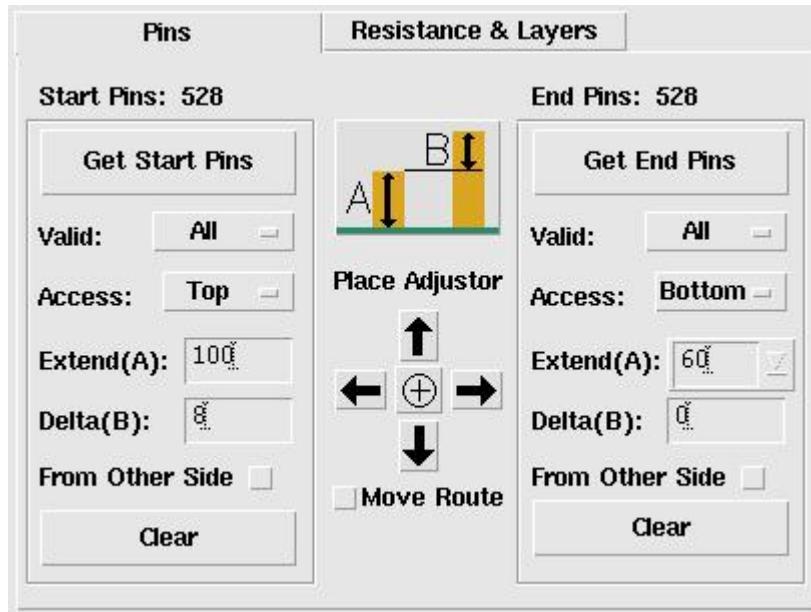


A light highlight line is drawn between each matched pair of start pin/end pin.



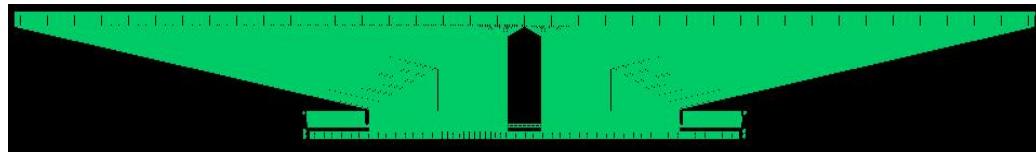
Set start/End pin extension (“**Extend(A)**”) and increment (“**Delta(B)**”). Start/End pins will be extended and the flight line is updated, and the first point of the enter points command is also update.

**Note:** Always set some pin extension (“**Extend(A)**”) for both start pins and end pins, it will result in bad routes easily when “**extend(A)**” is 0.



Click “**Preview**” or “**Auto Route**” to view the result (In FaceToFace structure, “**Preview**” and “**Auto Route**” is same). The program will connect the current routing edges to the end pins correspondingly and draw the result (not highlight) into the cellview. You can continue your routing, adjust points/width and/or delete routing points (BACKSPACE key) after “**Preview**”. If there are bad connections (those self-intersect, going from the back side of start pins, or going to the back side of end pins), they will be drawn in (“marker” “warning”) layer.

Enter “**ESC**” key after preview to accept the result or Click “**Finish Route**” to compute and draw the route.



Please refer to examples in rRouter(path) for handling wiggle patterns.

## 2) FaceToSide Structure

Start rRouter(polygon) command.

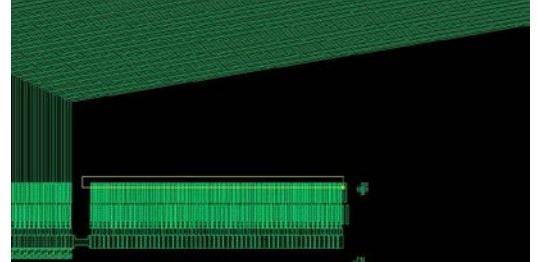
Set “**Target R Values**”: “**Auto**” and proper “**Min Width**”, “**Min Space**” etc rules.

Set “**Wave Pattern**” to “**Straight**”

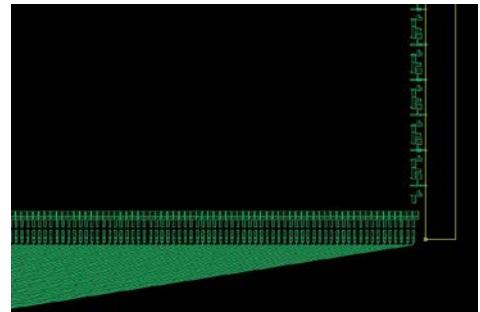
Click “**Get Start Pins**”, and draw a selection box over start pins (Driver Pins). The total number of start pins will be shown in the command form.

Make sure the pin “**Access**” direction is “**Top**” if the program does not automatically set it correctly.

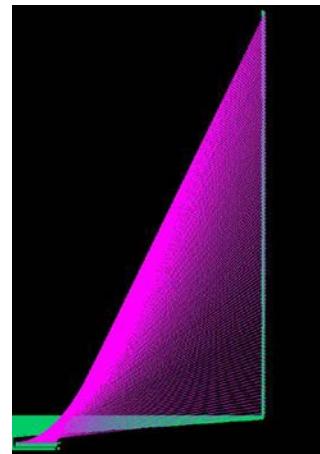
The “**No. of Route**” and “**Route Range**” are updated to the number of start pins and total range of start pins respectively, and the first routing point is snapped to the center of start pins.



Click “**Get End Pins**”, and draw a selection box over end pins (Flat Panel Pins). Make sure the pin “**Access**” direction is “**Right**” if the program does not automatically set it correctly.

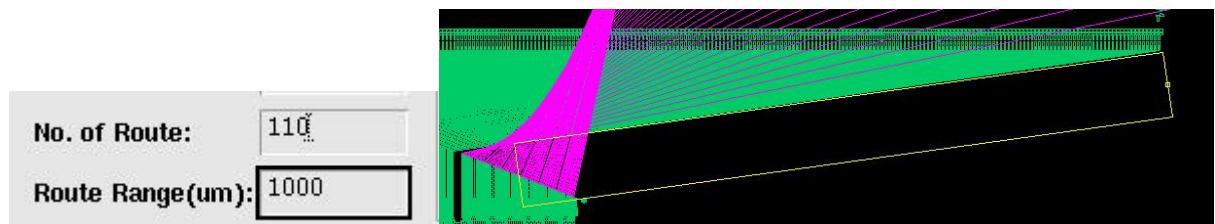


A flight highlight line is drawn between each matched pair of start pin/end pin.

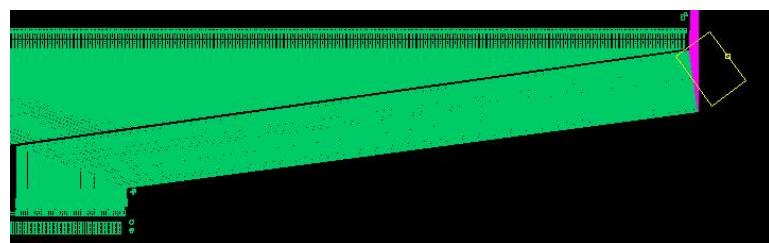


Set proper start/End pin extension (“**Extend(A)**”) and increment (“**Delta(B)**”). Start/End pins will be extended and the flight line is updated, and the first point of the enter points command is also update.

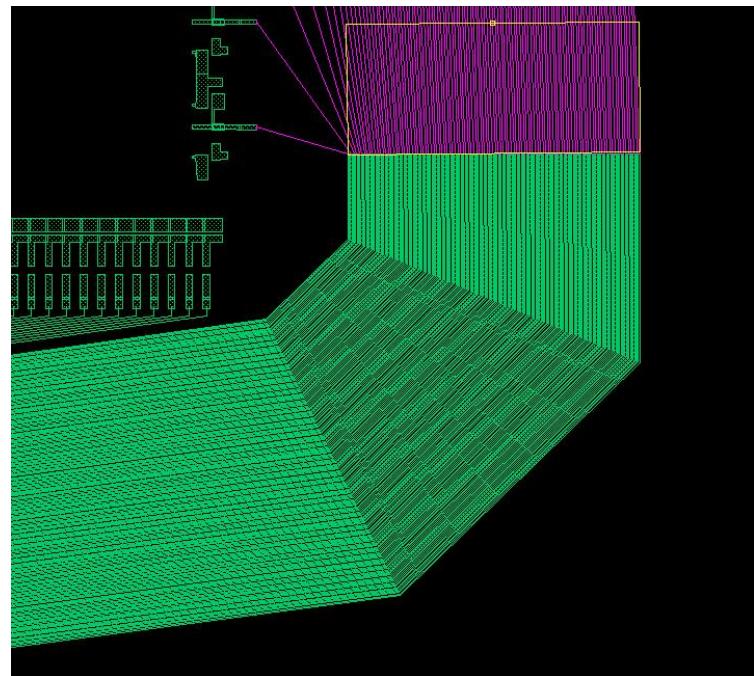
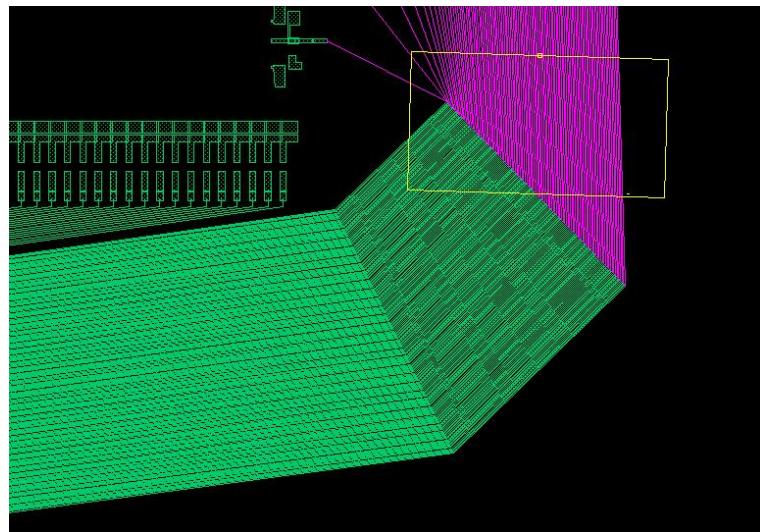
Change “Route Range” to a proper value, it will be the total route width for the next routing point you entered.



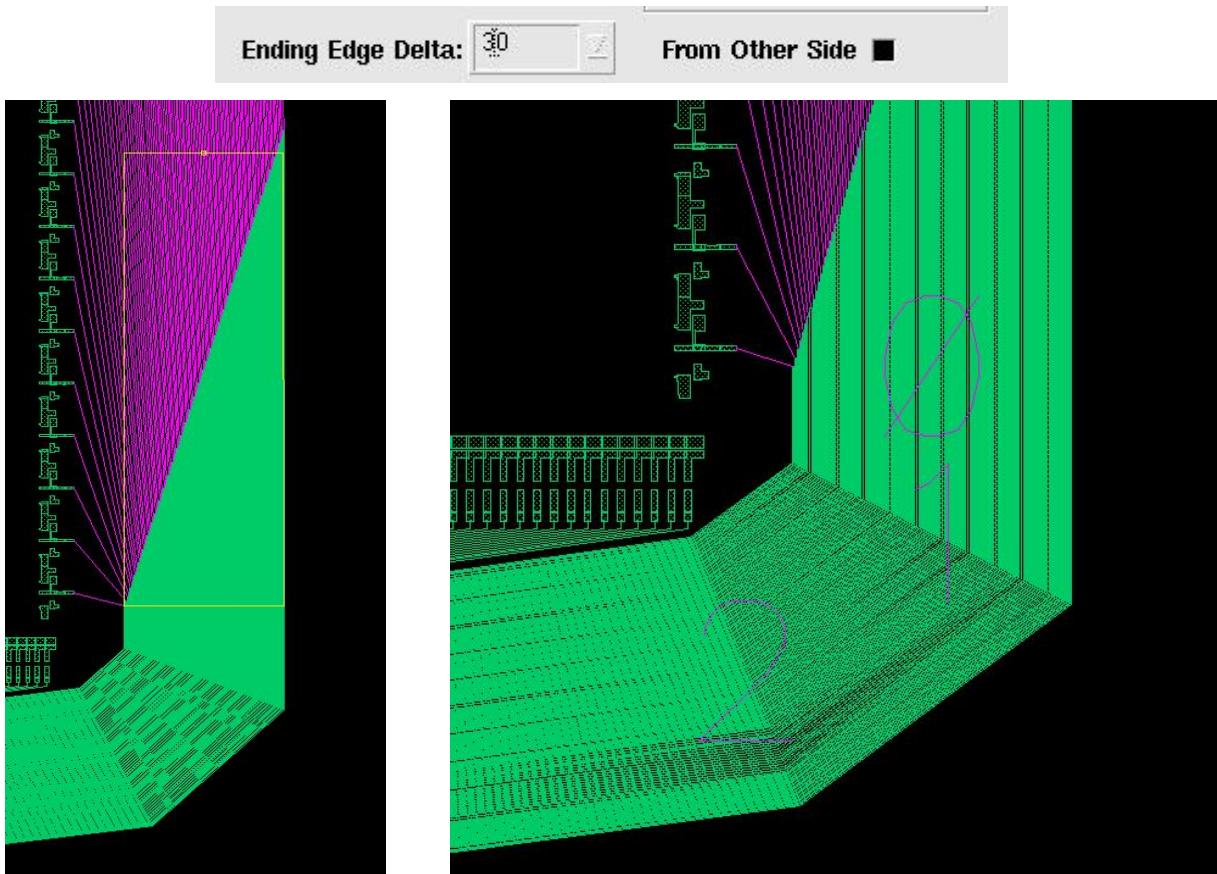
Enter Routing Point, and the wires are update on fly.



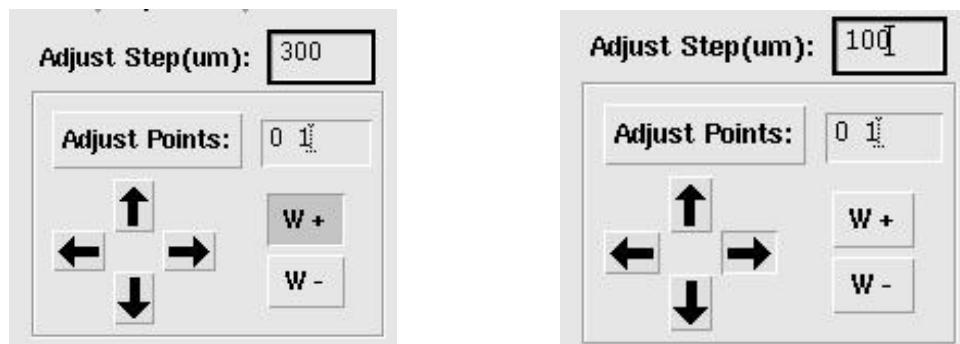
Continue entering routing points. There is no limit no number of routing points, or the angle of routing segments. As long as all routes can be created properly.



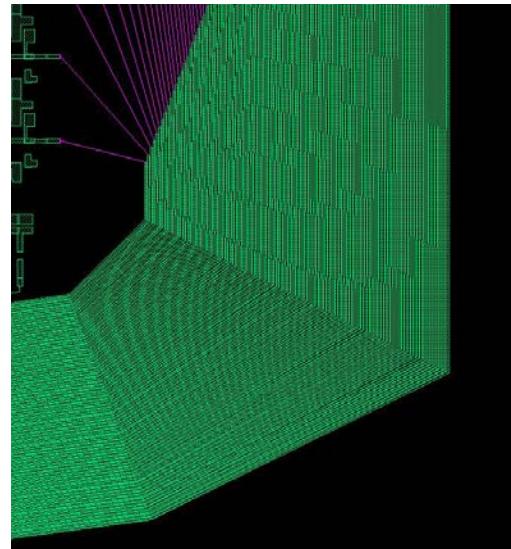
Adjust “**Ending Edge Delta**”, so that the ending edges of all routes will be extended properly for a better route result.



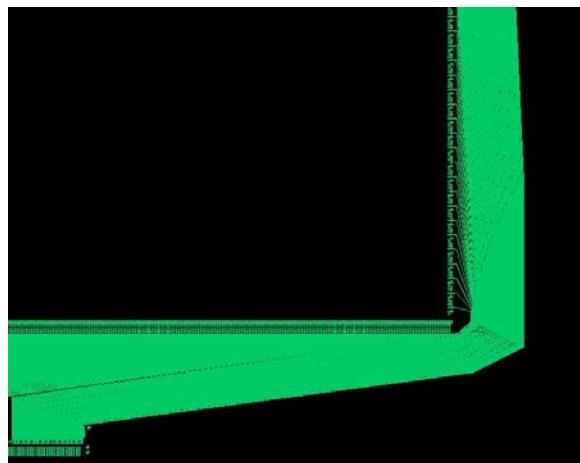
Click “**Adjust Points**” button, the entered routing points will be indexed (0 1 2...) from the last point. Input the points you want to adjust, and a proper “**Adjust Step**”. Click the arrow buttons to move the points. Click the “W+” / “W-” button to increase or decrease the routing width.



Increase the width at note 0 and 1 by 300um, and move point 0 and point1 rightward by 100um. The result will be update as show below.



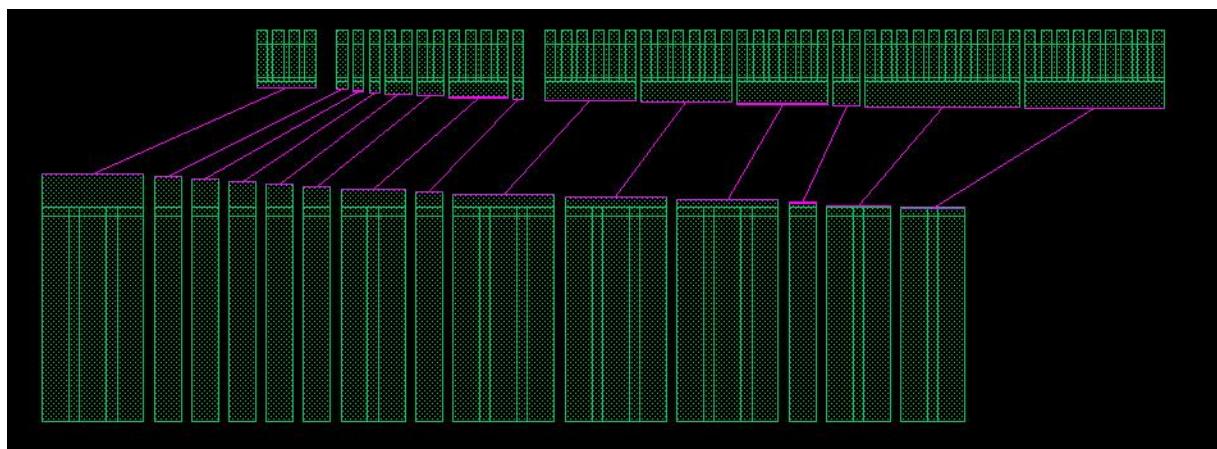
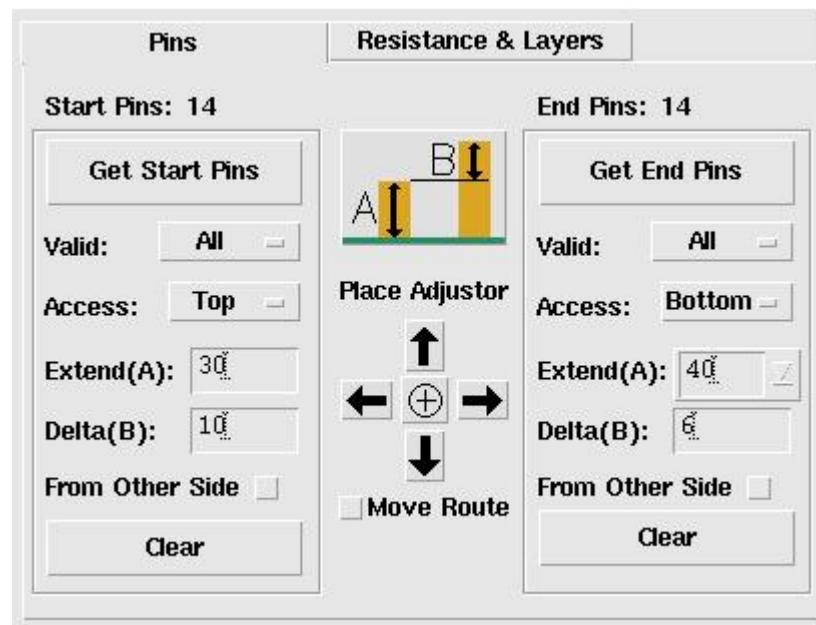
Preview and Finish Route



### 3) FPC connection

Select Pins and Set pin extension as in example 1.

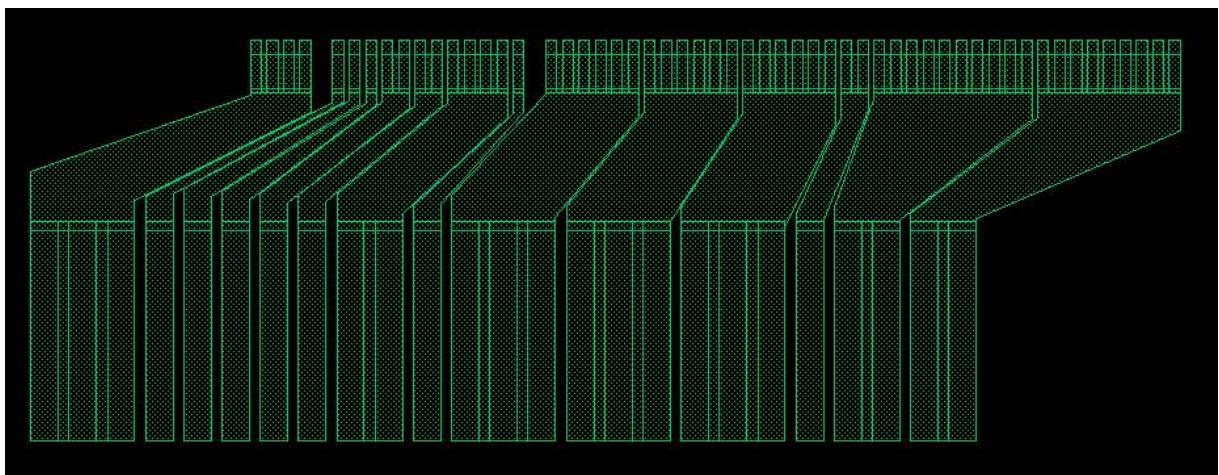
**Note:** Always set some pin extension (“Extend(A)”) for both start pins and end pins, it will result in bad routes easily when “extend(A)” is 0.



Select “Route Mode” = “Ratio W”

Route Mode: ◇ Equal R ◆ Ratio W ◇ Equal W ◇ Extender

Click “Preview” to check results. Increase Pin Extension if there are bad connections.



#### 4) R control

In this example, we want to create connections with specific resistance.

Set “Min Width” = 4, “Min Space” = 4, “No of Route”= 2 and “Routing Range” = 40.

Set “Target R Values” = 3. “Wave Pattern” = “Straight”

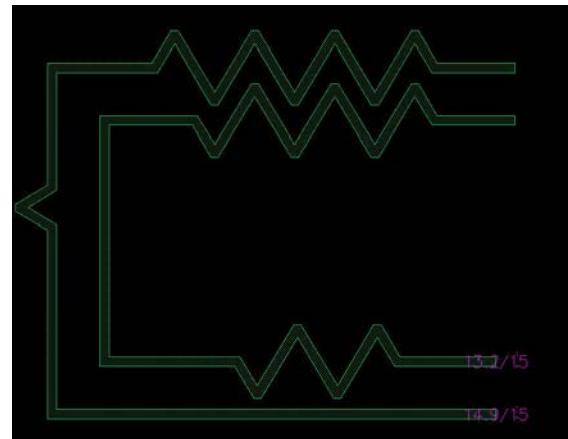
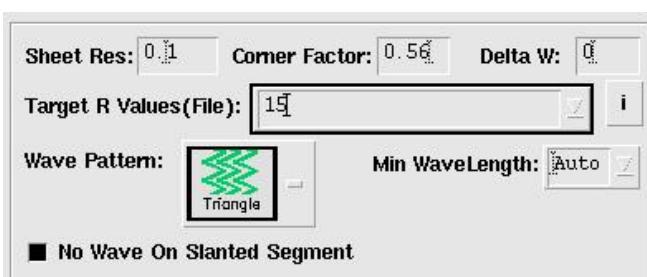
Pins		Resistance & Layers					
		Sheet Res:	0.1	Corner Factor:	0.56	Delta W:	0
Min Width(um):	4	Target R Values(File):	3				i
Min Space(um):	4	Wave Pattern:	Straight	Min WaveLength:	Auto		
No. of Route:	2	■ No Wave On Slanted Segment					
Route Range(um):	40						

Click to draw wires like below. The width of wires will change dynamically as the length of route changes. To meet the target R, the route width is minimum to maximize resistance when length is short. After it reaches the target value, the route width becomes wider to keep resistance equal (close) to the target value. The total Routing width will not exceed the given routing range(40) at each node. Resistance will increase if maximum allowed width is reached.

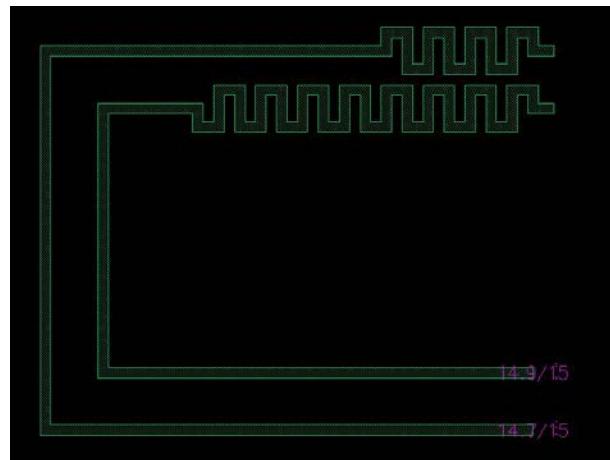
The width of the two routes is different and resistance of them is (close to) equal. The outer line is thicker than the inner one, because it is longer due to corner turns. The resistance labels at the end of line like “**3.01/3**” representing the actual resistance (3.01) of the wire and the target value (3).



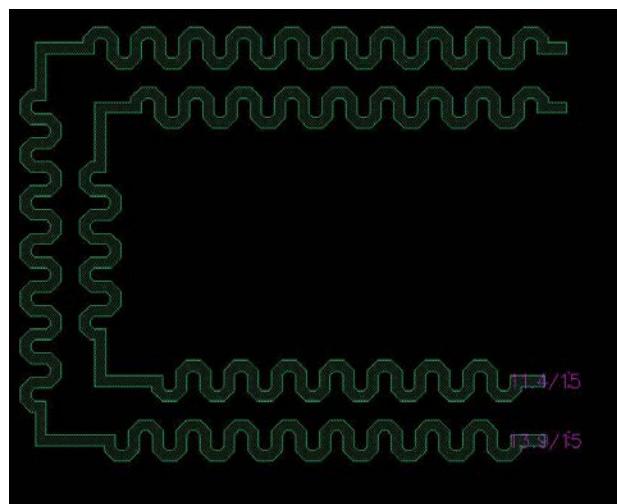
Now Change “Wave Pattern” to “Triangle” and “Target R values” = 15



Now Change “**Wave Pattern**” to “**Rectangle**” and keep “**Target R values**” = 15

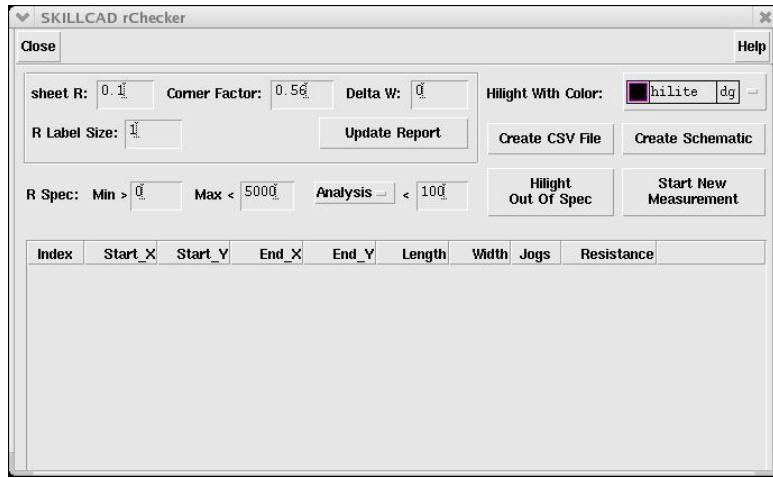


Now Change “**Wave Pattern**” to “**Octagon**” and keep “**Target R values**” = 15



## rChecker

rChecker can measure resistance for multiple of lines at one time, and works for both paths and polygons. The target shapes can be in lower or different hierarchy levels.



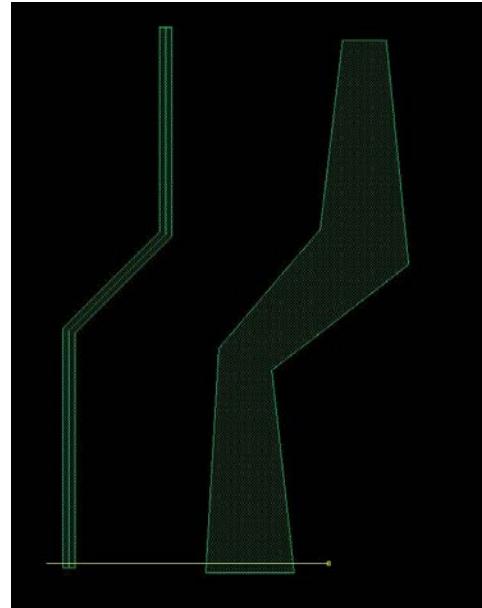
The program will use the following equation to calculate the resistance of a path:

$$\text{Path\_R} = \text{"Sheet R"} * \text{pathLength} / (\text{pathWidth} - \text{"Delta W"}) - (1 - \text{"Corner Factor"}) * \text{pathTurns}$$

To ignore the “**Corner Factor**”, set it to 1.0.

In case of a polygon shape, the shape will be tiled into trapezoids and triangles first; the program will calculate resistance for each trapezoid and triangle to get the total resistance of the polygon shape.

After you specify correct parameters, click “**Start New Measurement**”, and draw a line over the starting end of the shapes you want to measure resistance on. Target Shapes under the selection line will be highlighted with the layer specified in “**Hilight With Color**” option. You will be asked to pick one layer if there are multiple layers underneath the selection line.



Now draw a 2<sup>nd</sup> selection line across the other end of the highlighted shapes. Only shapes cut by both selection lines will be valid for resistance measurement.

The resistance of valid shapes will be listed in the report field of the rChecker form. Those shapes will be **indexed** by (1 2 3...). By default, the order is based on the positions of starting pins from left to right or bottom to top.

The following information are included in the report:

“Start\_X”/ “Start\_Y”: the center point of the starting edge.

“End\_X”/ “End\_Y”: the center point of the Ending edge.

“Length”: effective length of the shape

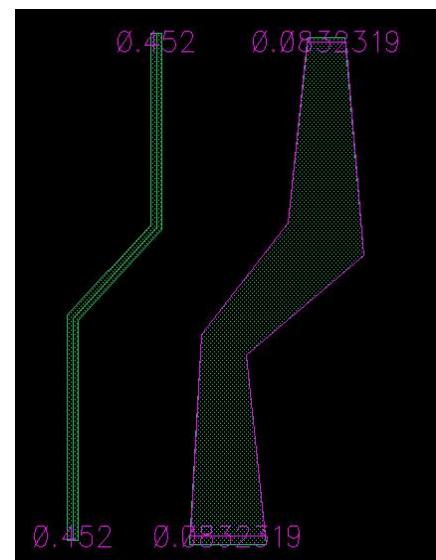
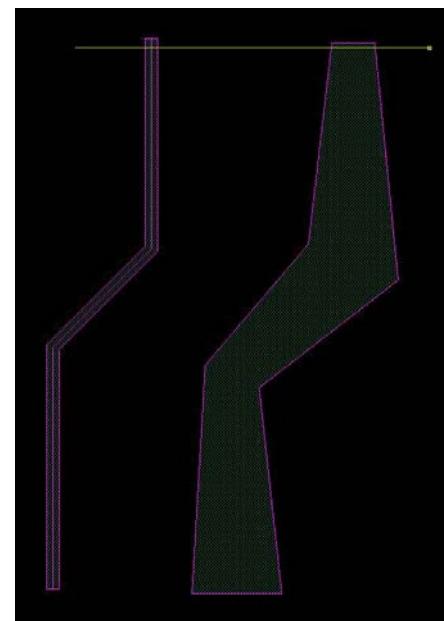
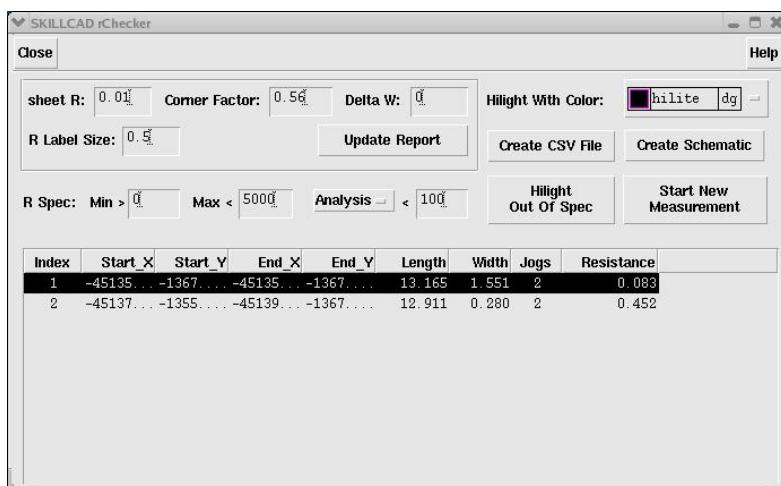
“Width”: effective width of the shape

“Jogs”: number of turns/corners.

“Resistance”: Resistance from starting edge to ending edge.

Click the header of a column; the result will be sorted according to that column.

Select one or multiple items in the report field, the corresponding shapes will be highlighted in the layout window. If “Sheet R” etc process info is changed, click “Update Report” the update the report without starting over.



Dev/Pre
R-Pre
Dev/Min
R-Min
Analysis

The program offers several ways to analyze the results.

“**Analysis**”: Default, no analysis will be done.

“**R-Min**”: The difference between the shape resistance (Shape R) and the minimum resistance (Minimum R) of all shapes in the report.

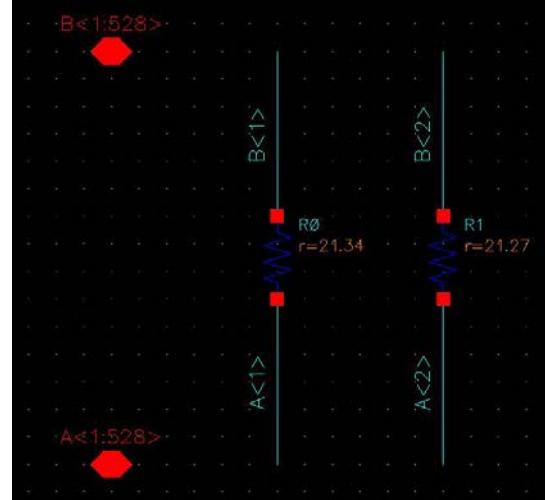
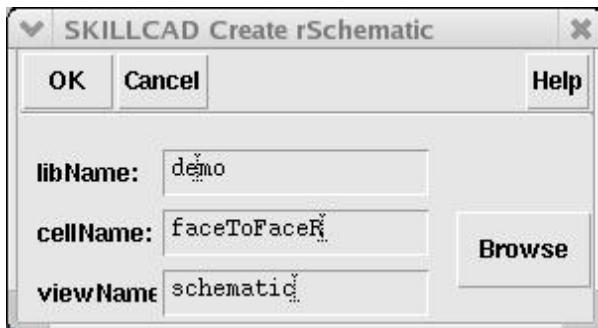
“**Dev/Min**”:  $(\text{Shape R} - \text{Minimum R}) / \text{Minimum R}$

“**R-Pre**”: The difference between the shape resistance (Shape R) and the resistance of the adjacent shape (Previous R). Make sure to sort report based pins position first.

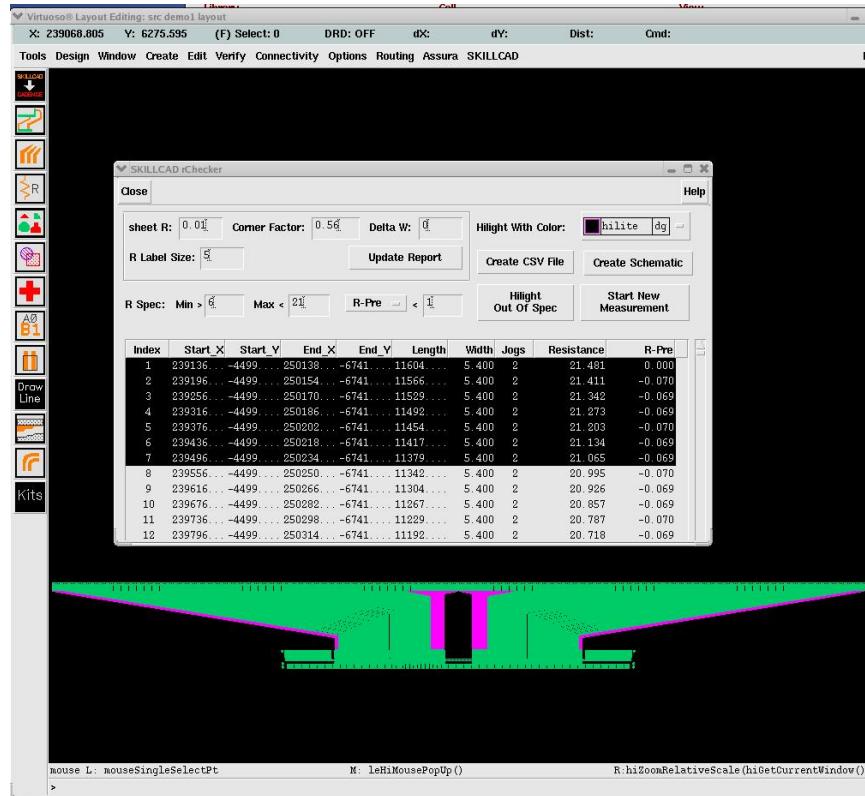
“**Dev/Min**”:  $(\text{Shape R} - \text{Previous R}) / \text{Previous R}$

The analysis result will be in a separate column in the report. The report can be exported as a CSV format file to third party data analysis tool by “**Create CSV File**”.

Click “**Create Schematic**” is mainly for post layout simulation. It will create a schematic view and symbol view for the connections. Each line of connections is represented by a parasitic resistor (presistor from analogLib) with the measured resistance value.

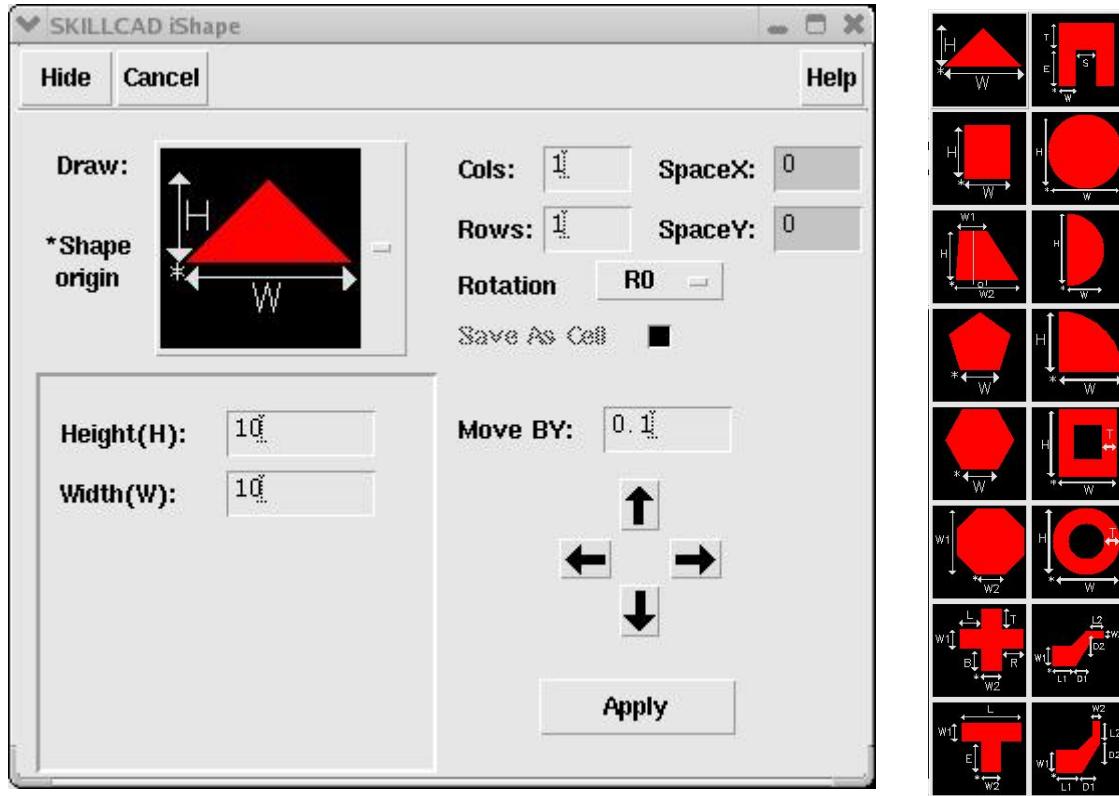


You can define a “**R Spec**” including “**Min R**”, “**Max R**” and the maximum analysis result. Click “**Hilight Out of Spec**” to select/highlight all routes do not meet the “**R Spec**”. You may need to take further action (modification) on those connections.



## iShape

This command is to create special shapes in plat panel layout design.



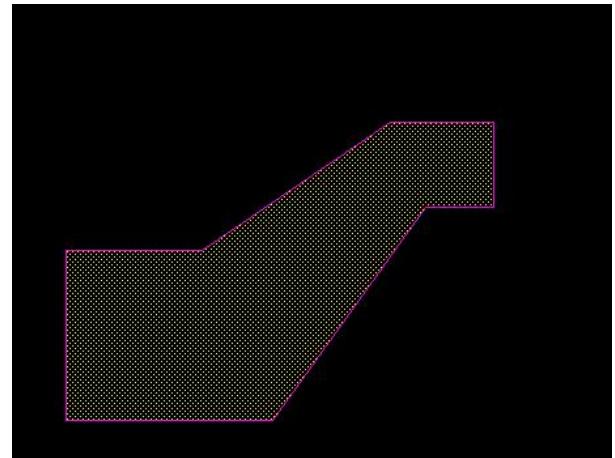
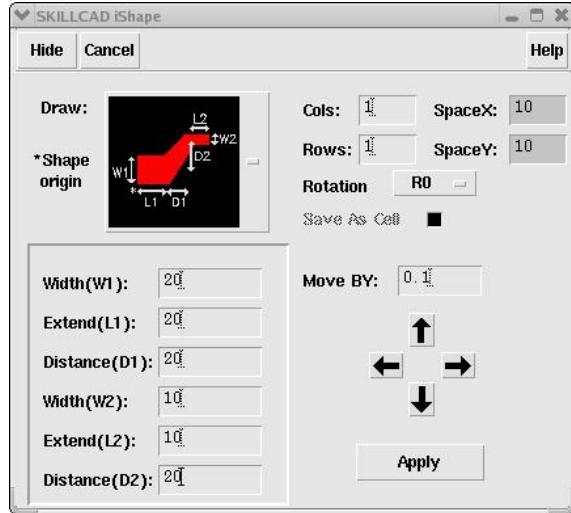
First select a shape you want to draw. The shape name will be printed in CIW (such as “draw Triangle”). The parameters of each shape are illustrated in the shape icon. The parameter field in the command form will be updated correspondingly. Symbol “\*” represents the origin of the shape.

Then click at the layout window to initialize the shape which can then modified by changing the values of its parameters including “Cols” “Rows” “SpaceX” and “SpaceY” array parameters. You can use the “array” buttons to move the placement. Click “Apply” to finalize the shape which will no longer be controlled by the command form.

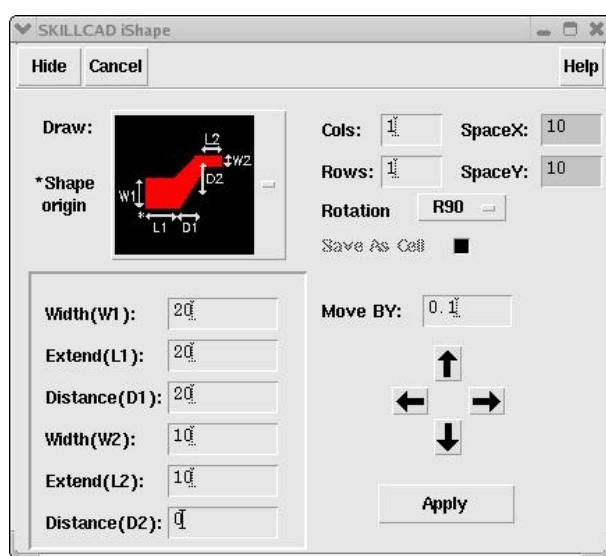
**Note:** Click at layout window will accept the current shape (if any) and place a new shape as defined in the command form.

## Example

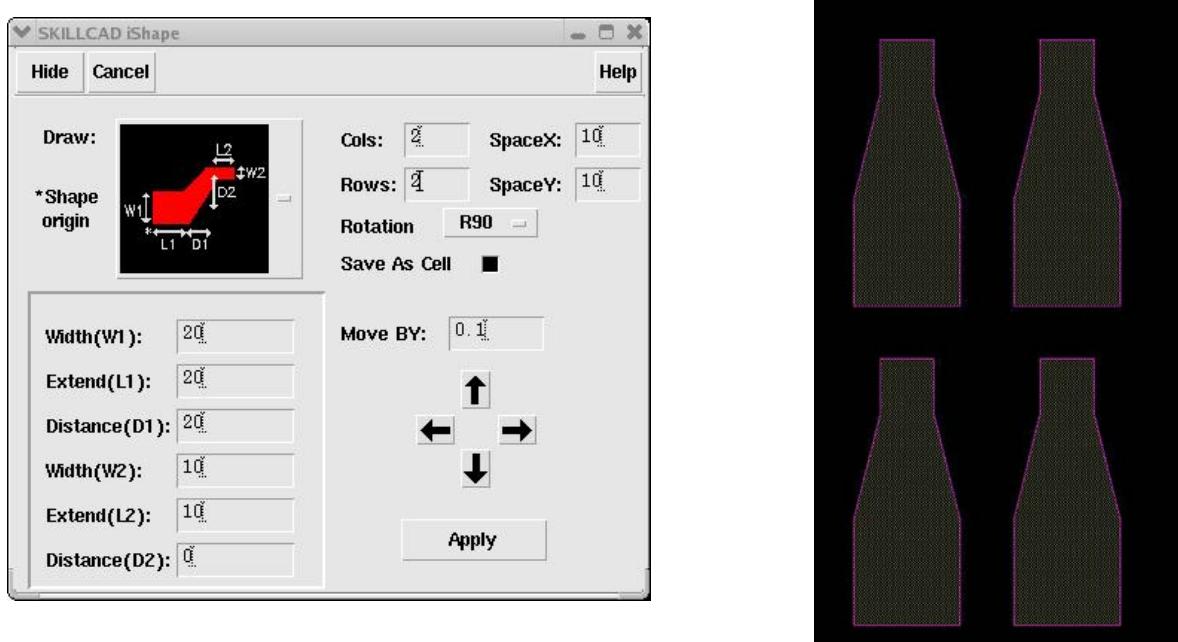
Pick the Shape (“zJoint”), and click at layout window to initialize a shape.



Set “**Distance(D2)**” = 0, and “**Rotation**” = R90.



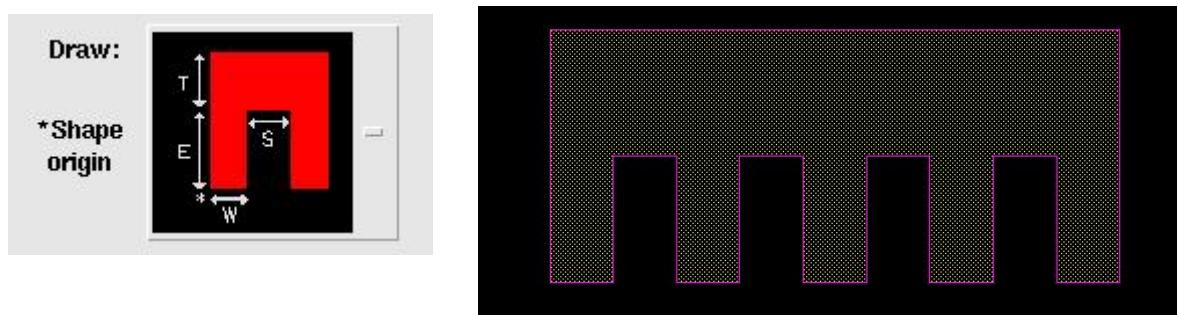
Set “**Cols**” = 2, “**Rows**” = 2, “**SpaceX**= 10 and “**SpaceY**” = 10



Click arrow buttons to move the placement.

Click “**Apply**”. You will be asked to input a cellname /group name (in IC6) for a shape array (Cols > 1 or Rows > 1) when “**Save As Cell**”/ “**Save As Group**” option is on.

You may have noticed that many new shapes can be created by changing the parameters of a shape. For example, you can create any number of fingers for “mPin” shape.



## Align

There are two alignment functions:

“Align One By One”

“Align All Selected”

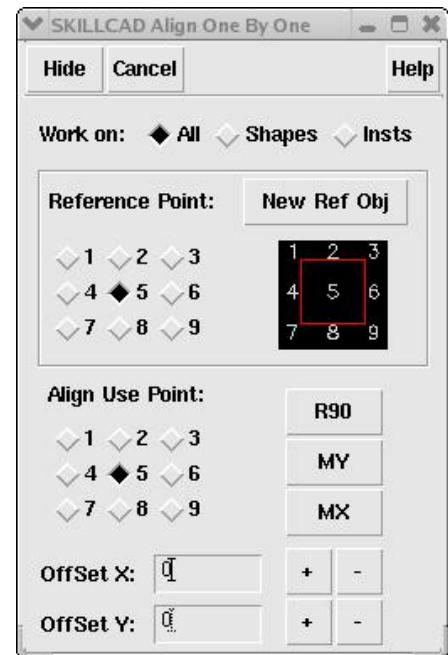


### Align One By One

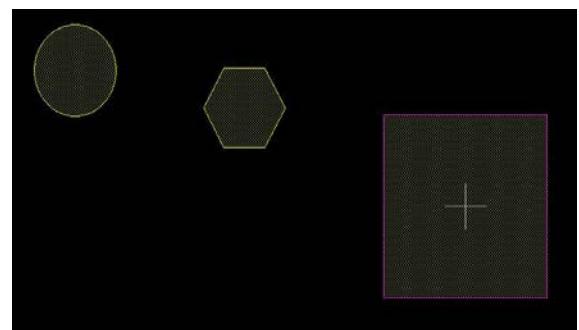
If an object is selected, the selected object will be automatically treated as “**Reference Object**” when this command is invoked. Otherwise, the program will ask you to click on an object to be the reference object.

There are nine reference points for an object as illustrated in the form. Click a radio button (1 – 9) to set the reference point for both reference object and the object to be aligned.

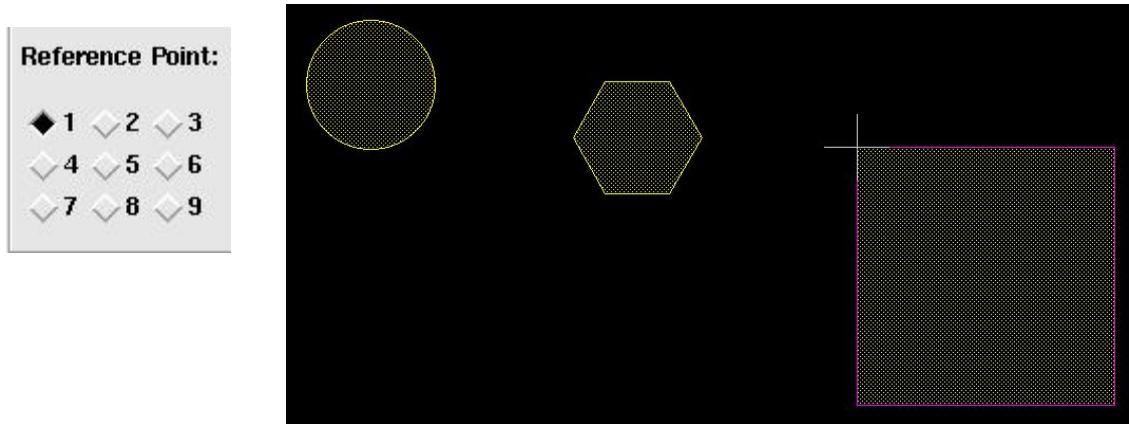
The aligned object will automatically become reference object for the next object.



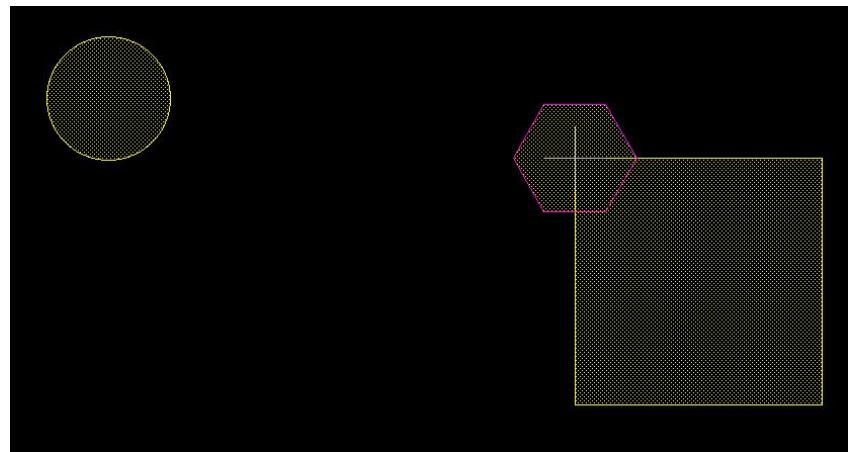
The example here for explanation has 3 shapes: a rectangle, a hexagon and a circle. First, click on the rectangle to make it as reference object. The object will be selected and The “**Reference Point**” (5) of it will be marked with “+”. And you are prompted to click on an object to be aligned now.



Change the reference point to “1” (top left corner), the “+” marker is updated immediately.



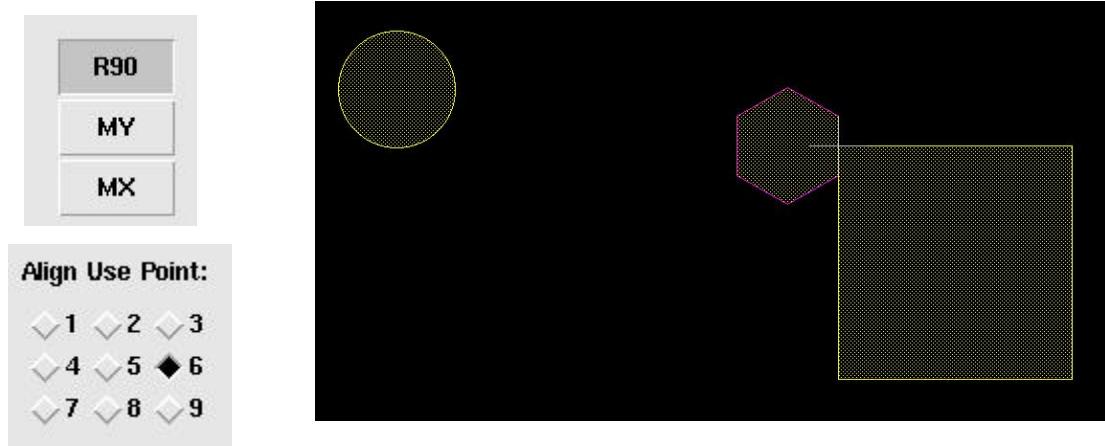
Click on the hexagon, the center (“**Align Use Point**” 5) of the hexagon is aligned to “+” marker. And the hexagon is now selected.



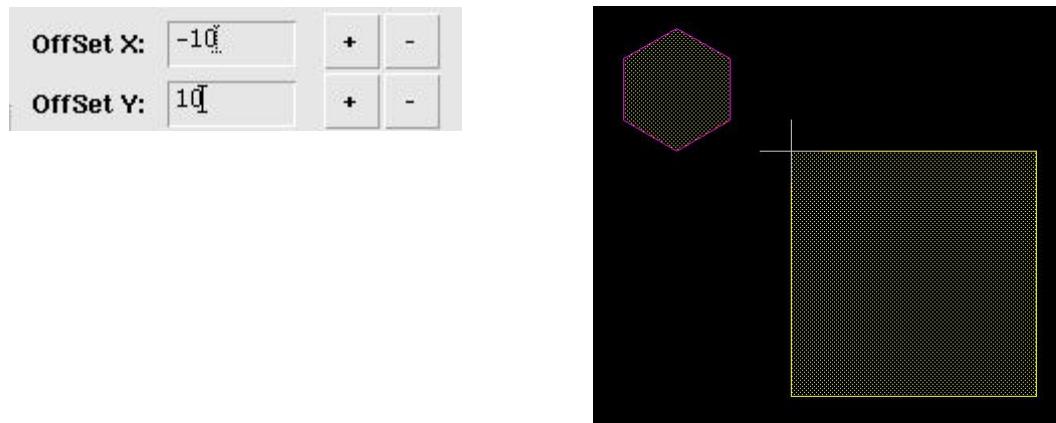
If multiple objects (shapes and instances) are underneath the selecting point, you will be prompted to select an object. You can use option “**Work on**” at the top of the command form to pick only “**shapes**” or “**Insts**”

Click “R90”, the hexagon is rotated by 90 degree (anti –clockwise).

And change the “Align use Point” = 6 (rightCenter).

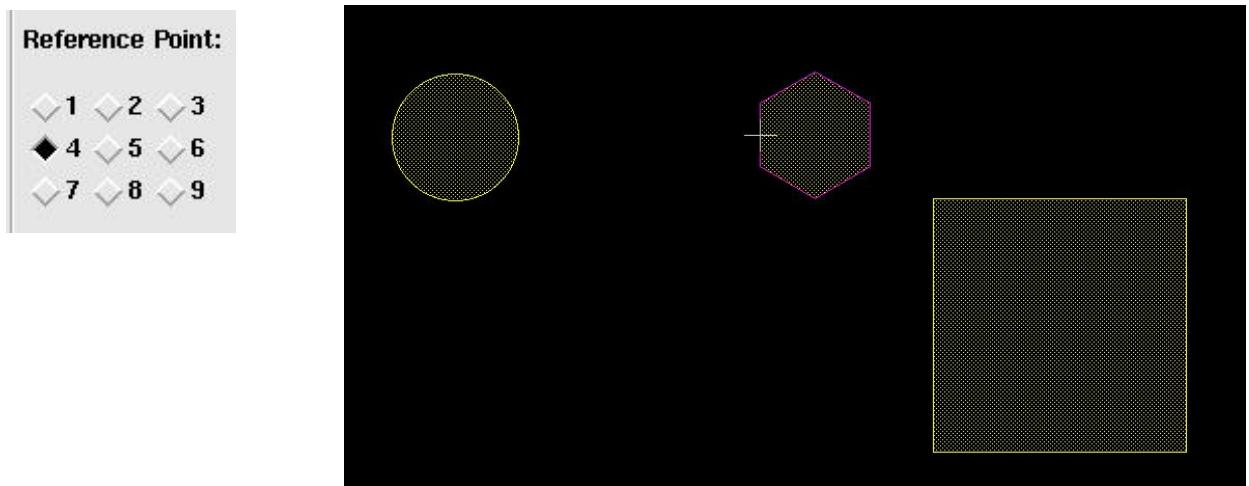


Set “Offset X” and “Offset Y” to move the current object. Click “+”/-” button will increase/decrease the offset by one layout grid.

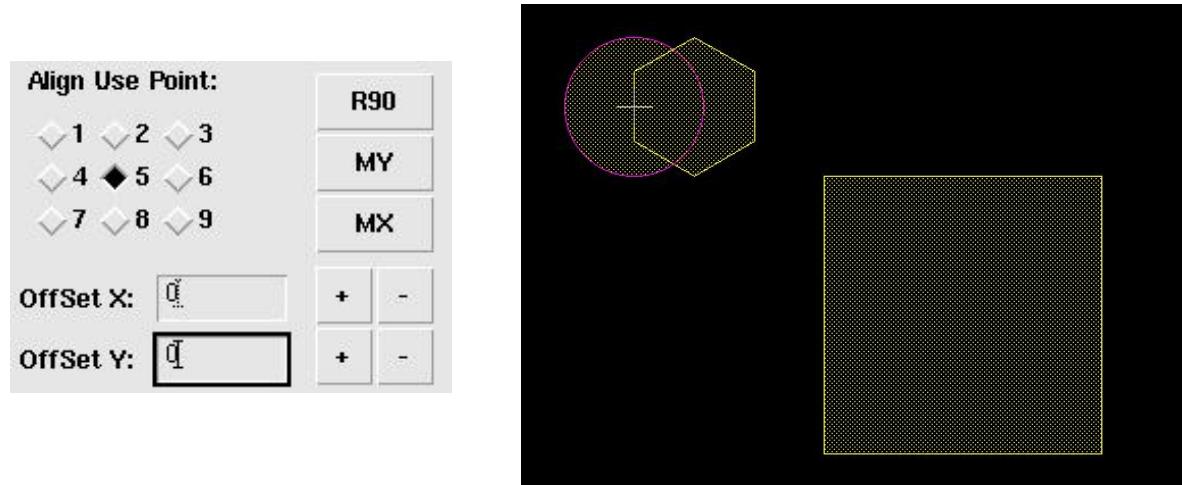


The aligned object will automatically become the reference object for next object to be aligned. Changing “**Reference Point**” or click on the next object will accept the current alignment (if any) and move the reference point to the current aligned object. Of course, you can always click “**New Ref Obj**” to select a new reference object.

Change the “Reference Point” = 4 (leftCenter), the “+” marker is on the hexagon now.



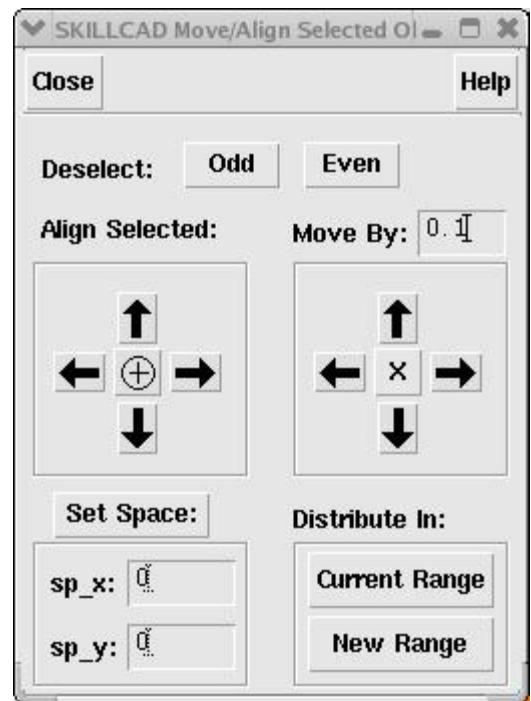
Click on the circle, and set “Align Use Point” = 5 (centetCenter),  
Offset X = 0 and offset Y = 0



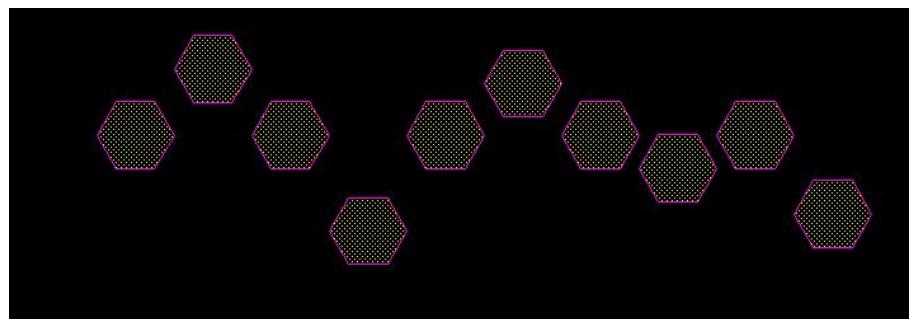
## Align All Selected

The command will align/move selected objects as a group.

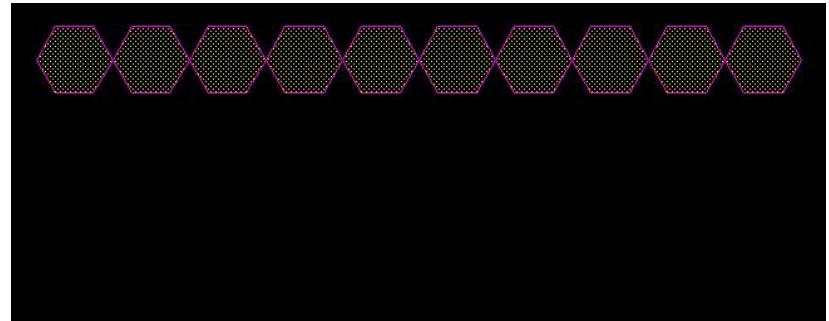
“Deselect” enables you to only work on Odd or Even objects based on the order in the selection group.



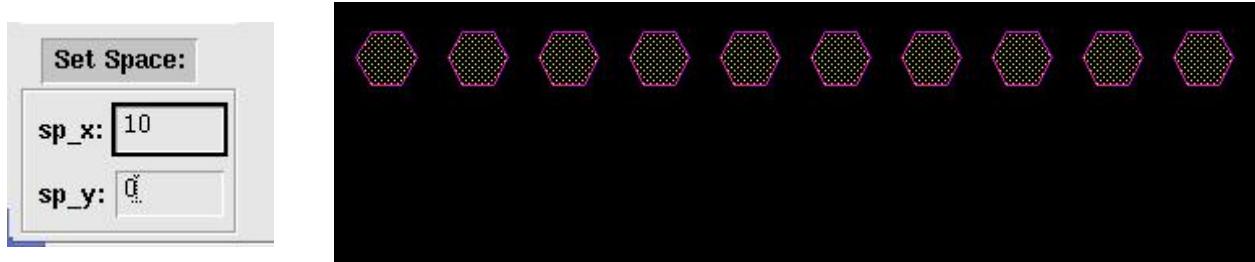
Original selected shapes .



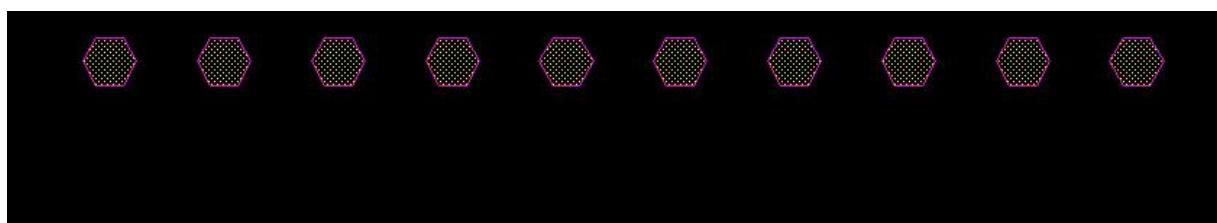
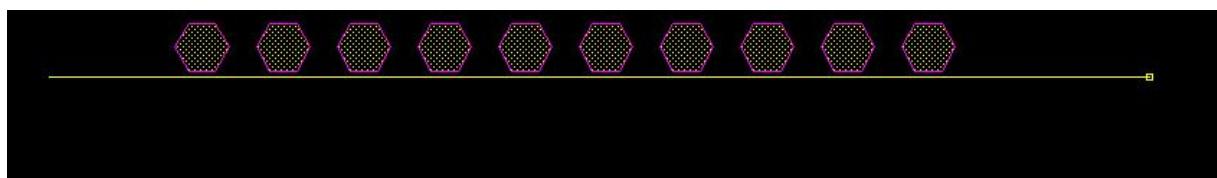
Align to “Top” . The center “+” button will align objects to longer center line of the group.



Set Space “sp\_x” = 10um.



Distribute the selected objects to a new range drawn by mouse.

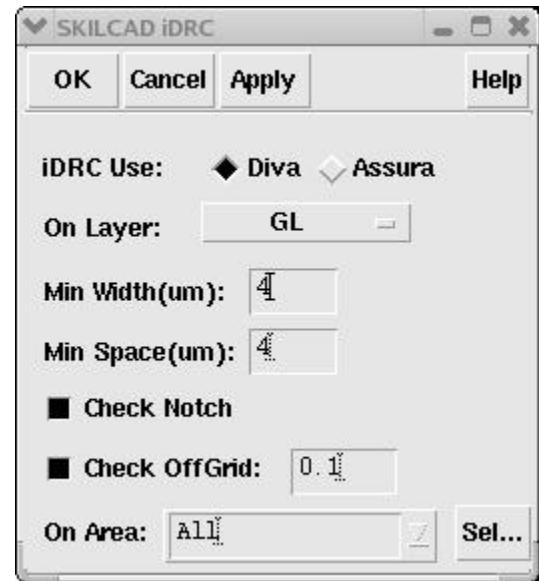


Use the move arrow button to move the group of objects by the amount specified in “Move By” to the corresponding direction. The “X” button will move objects by mouse.

## iDRC

The command will check simple DRC rules and offGrid on the specified layer with “**Diva**” or “**Assura**”. The command will create Diva or Assura deck on fly and execute it automatically.

You can also select an area (default is “**All**”) by click the “**Sel...**” button and draw a selection box over the layout cellview.



**Note:** A valid Diva or Assura DRC license is required to use this command. And an Assura DRC viewer license is required to view the assure DRC results.

## Mask Label

When “**Label Type**” is “**Mask Data**”, the command will create physical labels which can be seen in layout mask.

When “**Label Type**” is “**Text Display**”, the command will create normal labels which can be seen on in layout but will not be used in actual mask.

When “**From:To:Step**” is given, the command will create an array of labels like:

“**Label Name**” \$Index “**Suffix**”

Where \$Index will be replaced by actually number derived from the setting of “**From:To:Step**”

“**Label Pitch X**” / “**Pitch Y**” defines the center-to-center space between two adjacent member of the label array.

“**Font Space**” defines the space within one label.

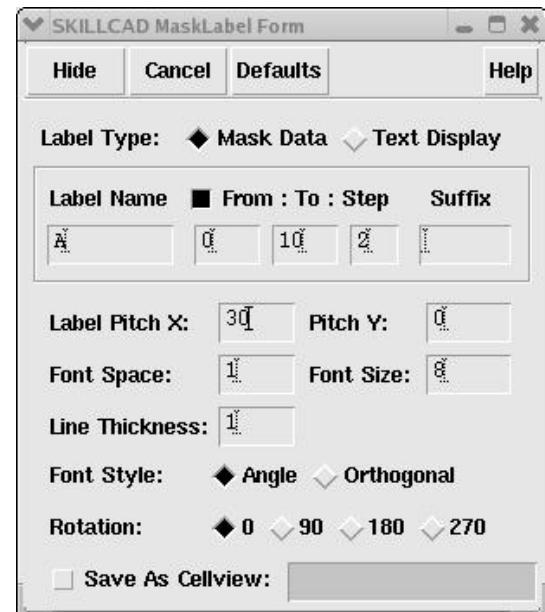
“**Font Size**” defines the height of the label.

“**Line Thickness**” defined the width of each line.

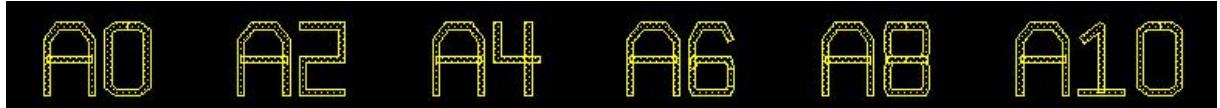
“**Font Style**” if is “**Orthogonal**”, all lines will be orthogonal.

“**Rotation**”: To rotate each label.

The label array can be save as a cell if “**Save As CellView**” option in checked.



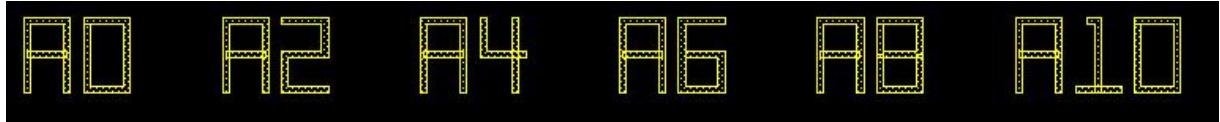
Example of Mask Label Array:



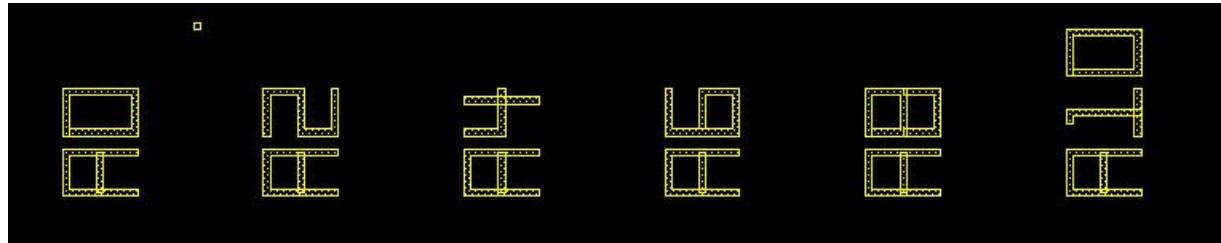
Bigger “Font size”, and thicker “Line Thickness”



Orthogonal Font Style.

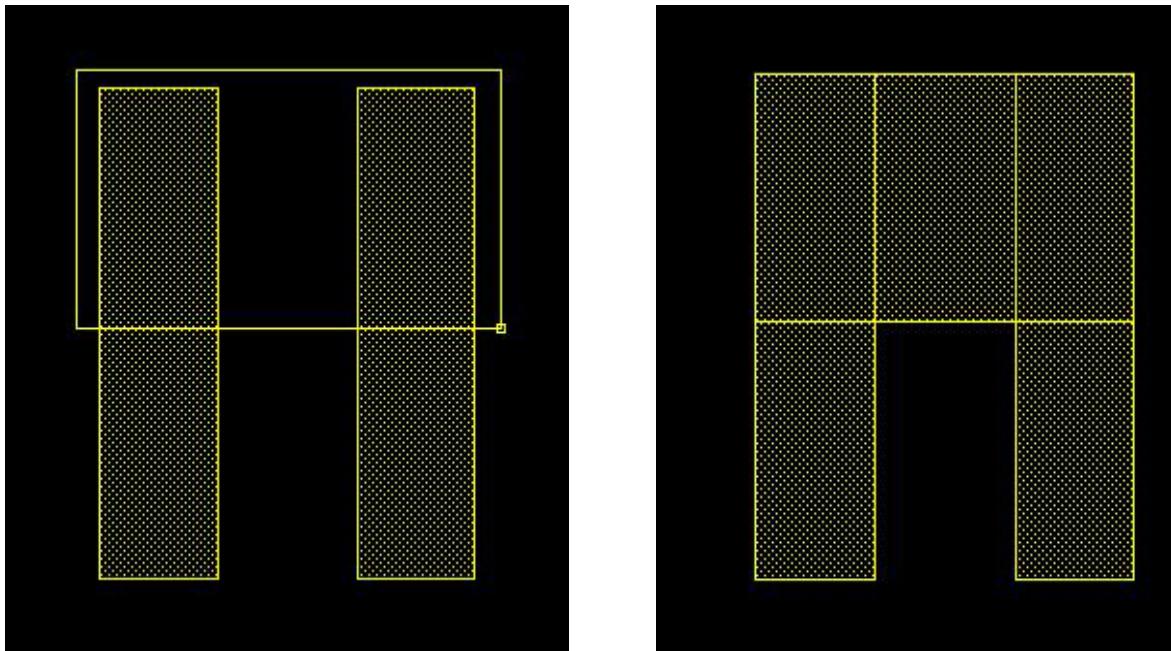


Orthogonal Font Style. Rotation “90”



## Merge Pins

Draw a box over pins to be merged. These pins will be joined by a rectangle which is aligned to the boundaries of original pins. The pins to be merged can be in lower different hierarchy levels.

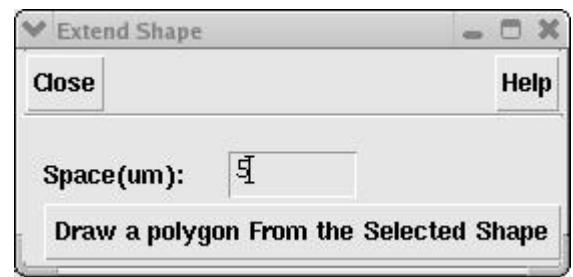


## **Draw Line**

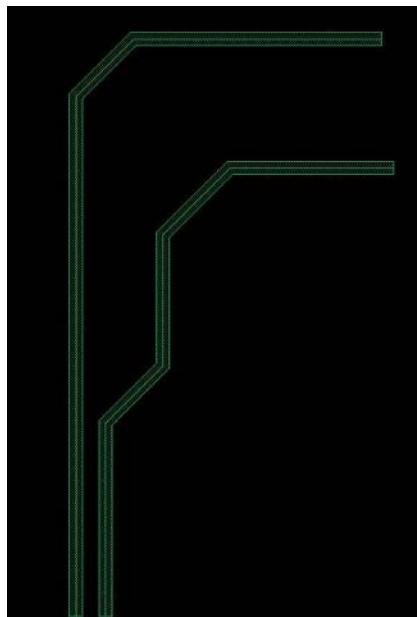
This function will create line on the current entry layer. The usage is similar as if you were drawing a path.

## Rule Extend

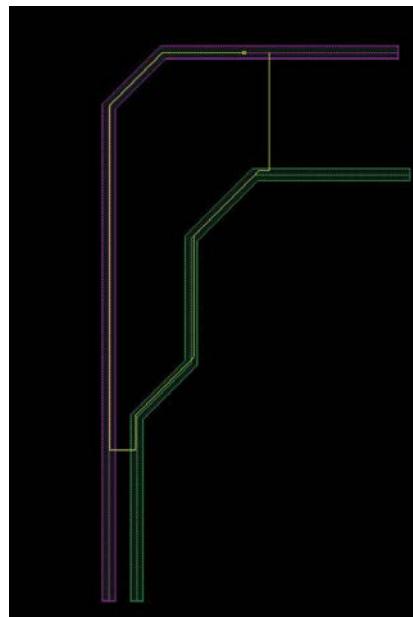
The function will fill gaps between shapes by extending the selected shape and keep proper distance from the outline of other shapes.



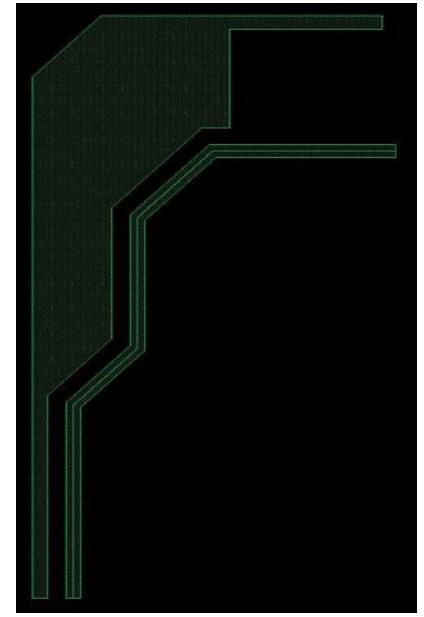
Select the object to be extended first, click "**Draw a polygon From the Selected Shape**" and draw a closed polygon.



Original shapes.



Select one shape (in red)  
And draw a polygon as shown  
in the yellow rubber line



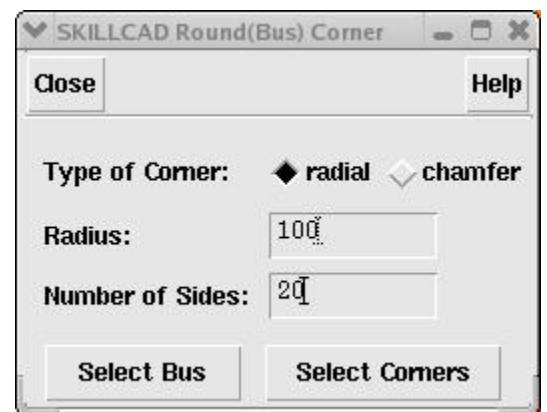
Result

## Round Corner

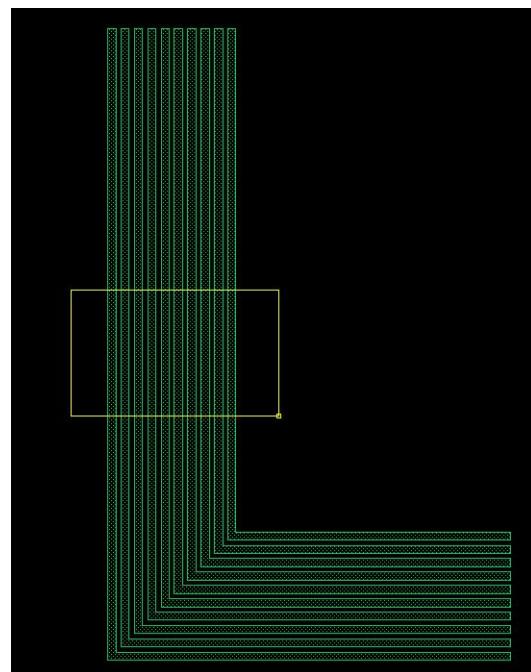
This function is to round bus corners in “**radial**” or “**chamfer**” style.

“**Radius**” determines the size of rounded corner.

“**Number of Sides**” determines the smoothness of the rounded corner (only applicable for “**Radial**” type)



Click “**Select Bus**” and draw a selection box to select a group of wires.

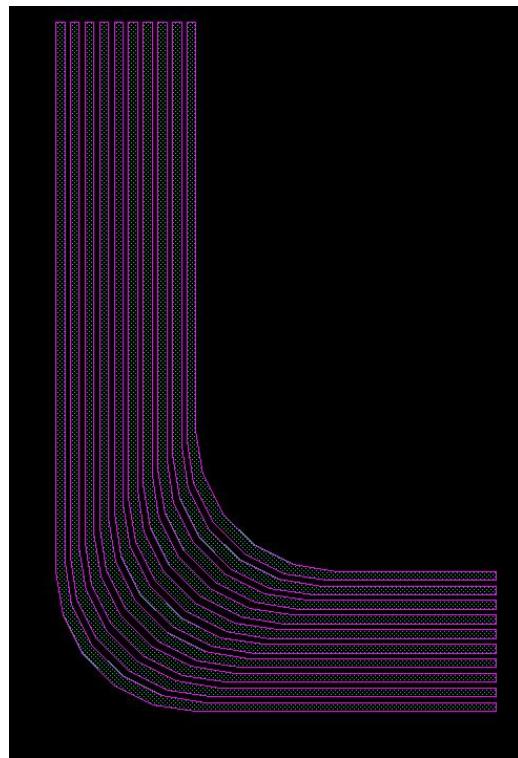


Click “Select Corners” and draw a selection box over the corners of the selected wires.

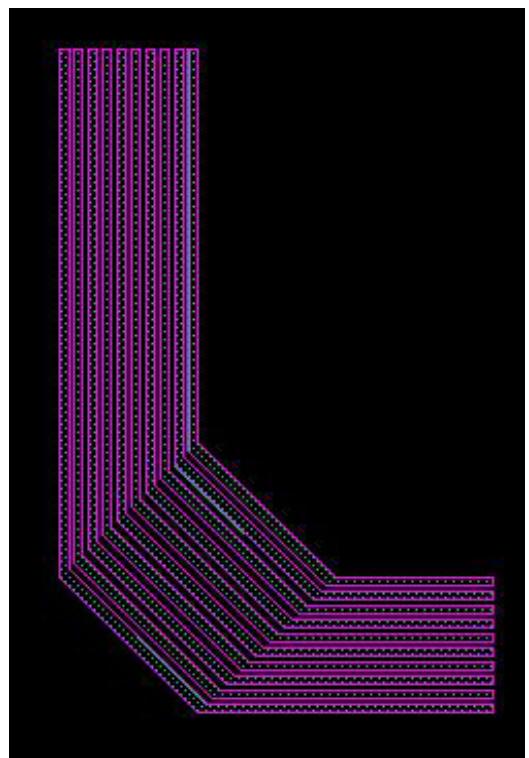


Result:

Radial Style



Chamber Style

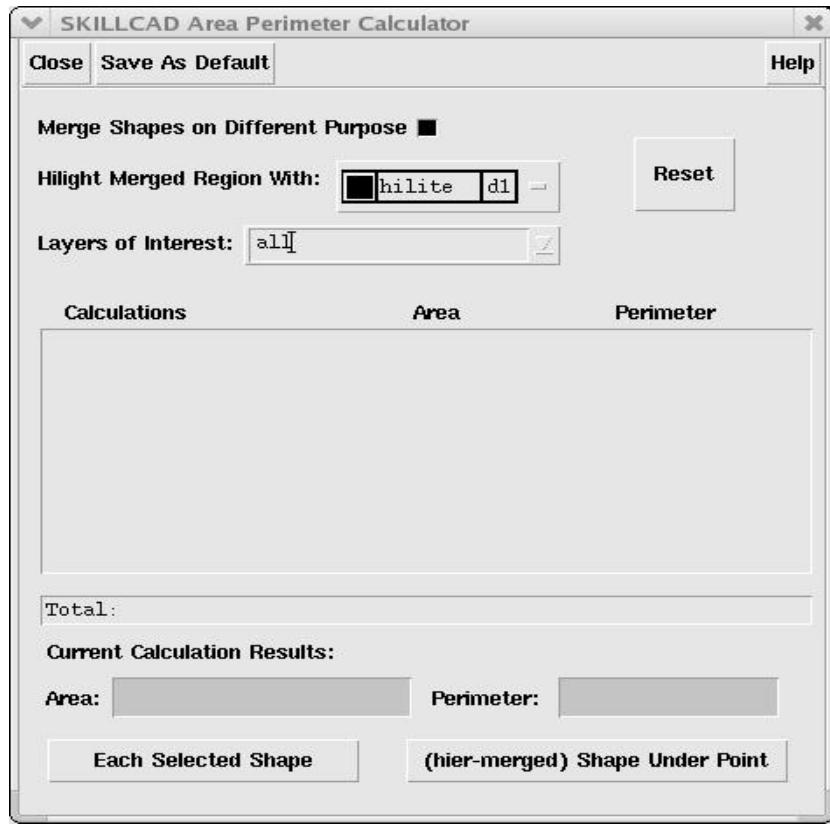


## KITs

Kits is a collection of small utilities.



## Area Perimeter Calculator



Two ways to use this function:

- 1). Shape Under Point (default).

As prompted, click on the shape (can be at lower hierarchical level) in the current layout window. The program will hierarchically extract the shapes on same layer and calculate the total area and perimeter as if these shapes are merged into one piece. The final merged shape will be highlighted (Just highlight, the layout data will not be changed) for your information. The highlights will automatically removed when this command form is closed.

When “**Merge Shapes on Different Purpose**” is on, shapes on same layer but different purpose, such as (“M1” “drawing”) and (“M1” “pin”), will be merged in this calculation.

The layer name, area value and perimeter value will be displayed in the center list box. You can continuously do multiple calculations, and each result will be displayed in the list box orderly. The program will sum up the result automatically.

When you select a measurement in the list box, the program will zoom to the corresponding region of the measured shape.

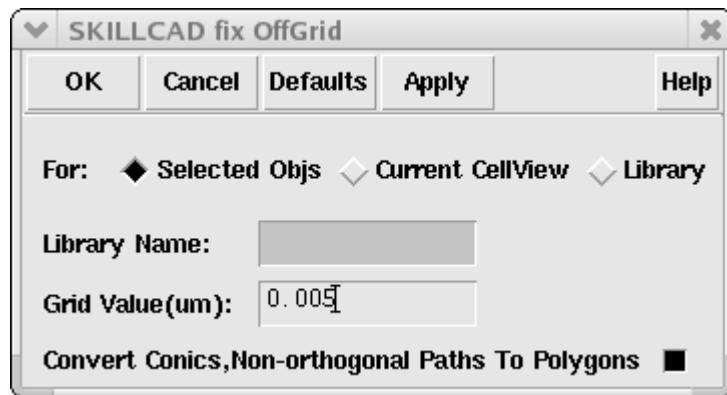
If more than one layer is under the clicking point, a form will pop up and asking you to pick your target layer. You can also preset your interested layer names (no purpose name) in the “**Layers of Interest**” field, and then only these layers can be picked for calculation.

## 2) Each Selected Shape.

In this mode, none of the options in the command form will be applied. The program will just calculated the Area/Perimeter for each selected shape and display the result in the list box.

“**Reset**” button will clear the list box and remove the total area/perimeter values.

## Fix Offgrid



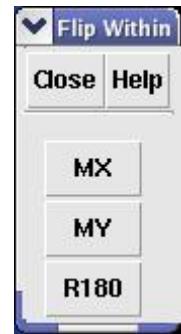
After you input the grid value, and library name (if fixing grid for an entire library), click the OK or Apply button, the program will start fixing offgrid for “**Selected Obj(s)**”, “**Current CellView**” or each layout cellview of “**Entire Library**”.

The current editing layout will not be saved automatically after fixing the grid for “**Selected Obj(s)**” or “**Current CellView**”. However, the program will save the layout view when doing for entire library, so remember to back up your data before running this program for a library.

45-degree paths will be converted to on-grid polygons if option “**Convert Conics, Non-orthogonal Paths to Polygon**” is checked. Otherwise, offgrid will not be fixed on these shapes.

## Filp within BBox

Filp (by MX, MY or R180) the select objects( shapes and/or instances) within individual original bounding box



## Full Selection

**"Partial To Full"**: partially selected shapes (if any) will be fully selected after clicking this button.

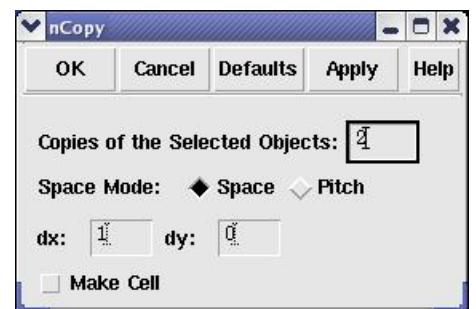
**"Full Select by Click"**: click to select entire shape.

**"Full Select(bus) By Box"**: select all bus by a selection box.



## nCopy

Copy selected objects n times based on given  
“Space”/ “Pitch”, and with option to save then as cell.



## Align/Distribute Pins(objs)

Refer to the align section for details. This function may be removed from Kits since now it can be invoked from Align function.