

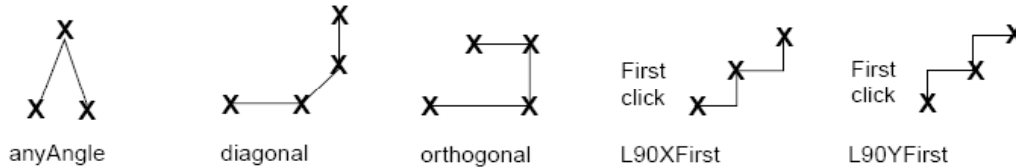
## 1. 关于版图一些实用的快捷键

**F3:** 显示Option form

**F4:** Full/Partial 选择切换

**N:** 改变snap model, n---diagonal, Shift+n---orthogonal, Ctrl+n---L90Xfirst

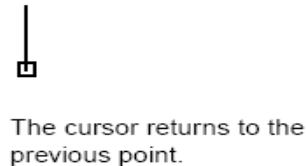
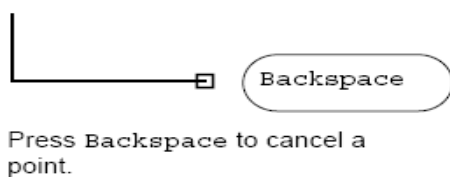
*Snap Mode examples:*



To toggle between *L90XFirst* and *L90YFirst* while you are creating a path, click right.

**Ctrl+y:** 当多个图形叠在一起时（点击左键默认是两个图形间切换），可以轮流选择重叠的图形

**BackSpace:** 当命令尚未完成时，可以撤销上一次（多次点击可撤销多次）鼠标的点击。如：画path时可撤销前面鼠标错误的点击，选择很多图形stretch，点了reference point发现有多选，可撤销点击，去掉多选图形后再stretch。  
press Backspace.



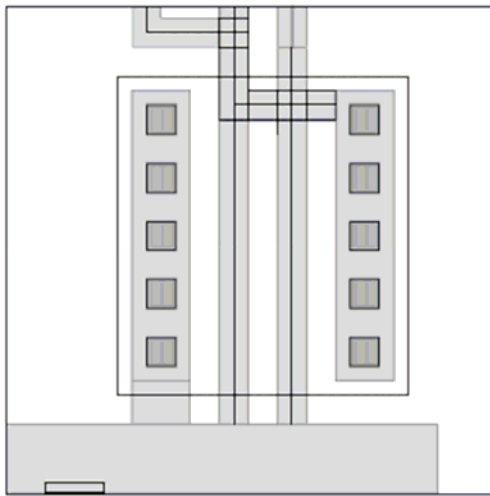
### Right mouse:

- 没有命令时重复上次命令；
- move和Create instance时逆时针旋转，**Shift+Right mouse** 轮流关于x/y轴对称；
- 画path时，L90Xfirst和L90Yfirst之间切换，**Ctrl+Right mouse** Path自动换层（Path stitching）切换，**Shift+Right mouse**换层时通孔旋转；
- Reshape和split时，切换不同的高亮区域，以便下一步的操作。

## 2.使用reference window

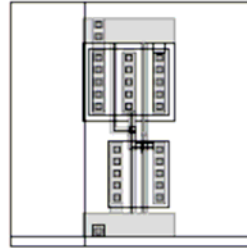
一个cellview可以打开两个窗口，一个作为主窗口编辑，另外一个可以放小一点作为参考窗口（即reference window），有点像world view，不同的是主窗口的编辑不仅在参考窗口中可以看到，而且两个窗口中编辑是等效的（当然你的显示器越大，用参考窗口越好，^\_^）。

可以用**Window - Utilities - Copy Window**打开一个参考窗口，也可以直接把一个cellview打开两次，如图



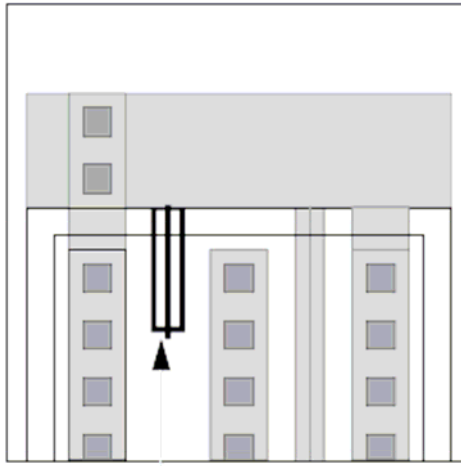
The original cellview window.

*Fit All* displays the entire design in the reference window.

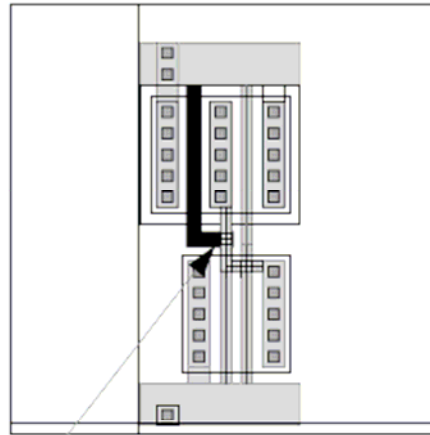


The new reference window.

可以同时两个窗口中编辑



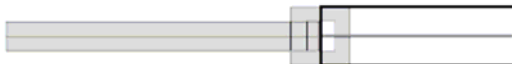
Start creating the path in the original window.



Finish the path in the reference window.

### 3.关于Path stitching

①画path时可以从一层切换到另一层，并且自动打上对应的接触孔，这个功能叫*path stitching*。



While path stitching, you can automatically place a contact and switch to another layer.

②在 *Change To Layer* 栏里选择你要换的layer，也可以通过 *Control+right mouse* 键来选择需要换的层。

如果 *Change To Layer* 栏里没有层可选，那是因为在 *technology file* 中没有关于这层的 *contact* (或者 *cdsVia* 的定义) 定义，只有定义了之后才能使用 *path stitching* 功能。

下面是一个 *contacts* 定义的例子：

```
*****
;
; DEVICES
*****
;
devices(
tcCreateCDSDeviceClass()

symContactDevice(
; (name viaLayer viaPurpose layer1 purpose1 layer2 purpose2
; w l (row column xPitch yPitch xBias yBias) encByLayer1 encByLayer2 legalRegion)

    (ACT_M1 CON drawing ACT drawing M1 drawing
    0.30 0.30 (1 1 0.60 0.60 center center) 0.14 0.15 _NA_)

    (GP_M1 CON drawing GP drawing M1 drawing
    0.30 0.30 (1 1 0.60 0.60 center center) 0.14 0.15 _NA_)

    (M1_M2 V1 drawing M1 drawing M2 drawing
    0.36 0.36 (1 1 0.71 0.71 center center) 0.12 0.12 _NA_)

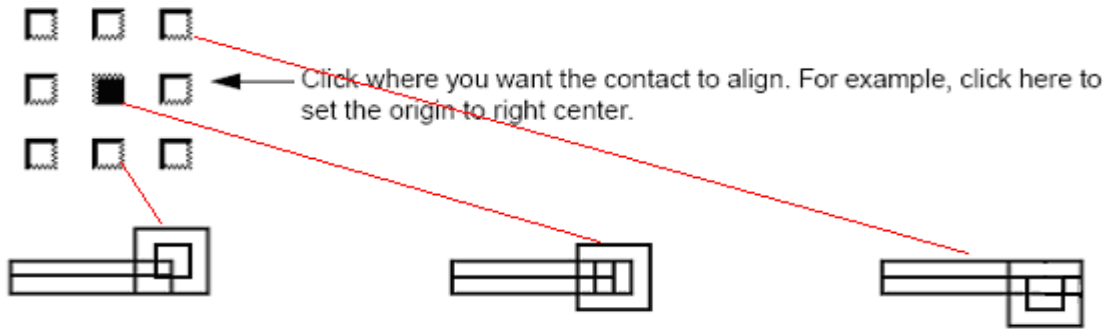
    (M2_M3 V2 drawing M2 drawing M3 drawing
    0.36 0.36 (1 1 0.71 0.71 center center) 0.12 0.12 _NA_)
); end of symContactDevice
..***** or you can define cdsViaDevice *****
;;
;cdsViaDevice(
;;( deviceName cutLayer cutPurpose layer1 purpose1 layer2 purpose2
;; row column origin stackedVias cutLayerW cutLayerL xCutSpacing yCutSpacing
;; layer1XEnc layer1YEnc layer2XEnc layer2YEnc layer1Dir layer2Dir )
; ( CDSVIA1 V1 drawing M1 drawing M2 drawing
; 1 1 centerCenter _NA_ 0.36 0.36 0.35 0.35 0.12 0.12 0.12 0.12 "" "" )
;
;
; ( CDSVIA2 V2 drawing M2 drawing M3 drawing
; 1 1 centerCenter _NA_ 0.36 0.36 0.35 0.35 0.12 0.12 0.12 0.12 "" "" )
;
;
; ( CDSVIA3 V3 drawing M3 drawing M4 drawing
; 1 1 centerCenter _NA_ 0.36 0.36 0.35 0.35 0.12 0.12 0.12 0.12 "" "" )
;
; ) ; end of cdsViaDevice
);devices
```

③根据 *path width* 的不同，工具会自动打上一个孔，或者是一组孔



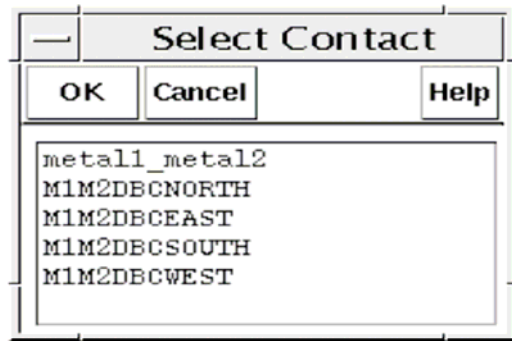
④如果想改变孔的对齐方式，可以通过改变 *Contact Justification* 来调整

## Contact Justification

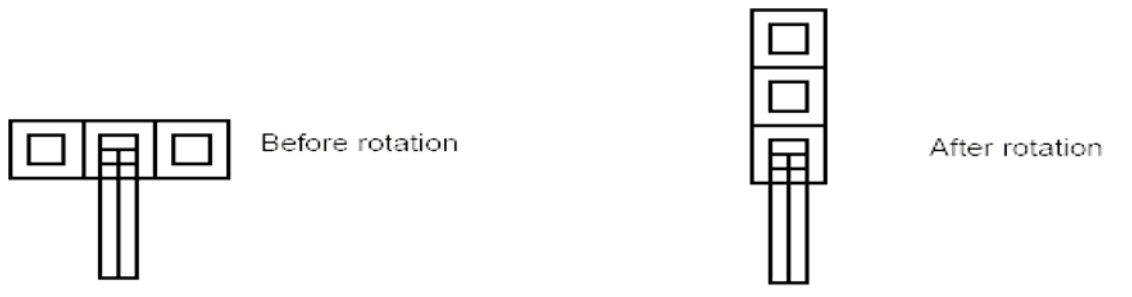


⑥如果定义了多个Contacts，Path Stitching时，

- 设置环境变量`useDefaultVia` 为`nil`，  
`envSetVal("layout" "useDefaultVia" 'boolean nil)`，
- 然后换层时，如果定义了多种contact，会弹出一个选择框，可以选择你需要的孔



⑦通孔只出现一个边框时，可以点击`Rotate` 或者`Shift and click right` 来旋转通孔。

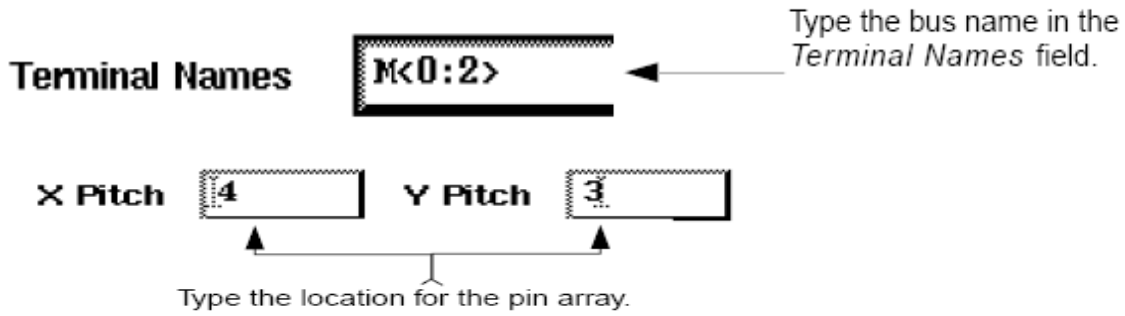


## 4.Placing Pin Arrays(bus pins)

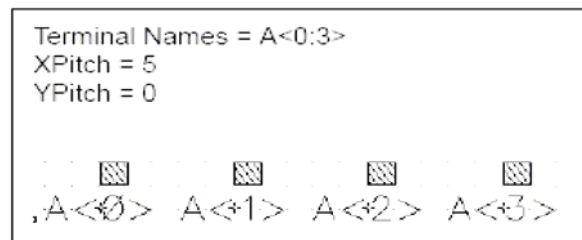
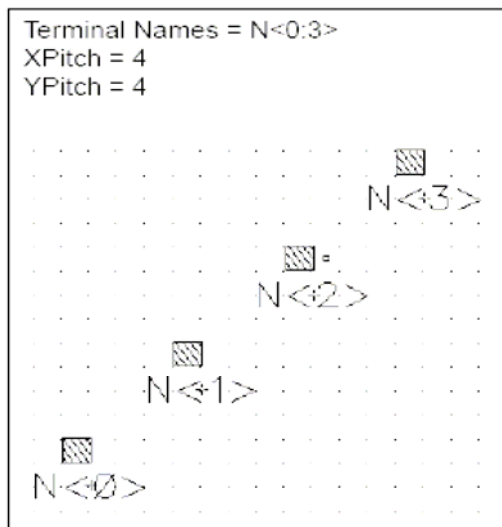
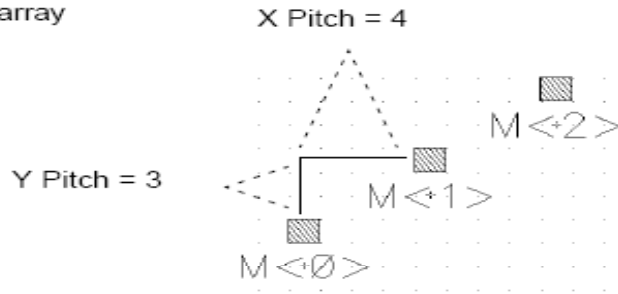
你也许认为label就可以代替pin了，还用的着打pin吗？其实不然，virtuoso工具一些连接关系都是通过pin来体现的，通过pin体现连接关系，在连线时（尤其时连线比较远，比较复杂时）可以通过打开Options->display->Display Controls->Nets，来显示复杂Net的连接。当然更实用的是在Layout-XL工具中可以对版图和电路进行实时对比（需要tech file的支持）。

当需要打bus pins的时候，可以一次打一组pins，bus pins的命名必须是Busname <startNum:EndNum>，这时可以通过修改Xpitch和Ypitch来改变Bus pin的间隔，注意打

的pin是顺序上升或者下降的。

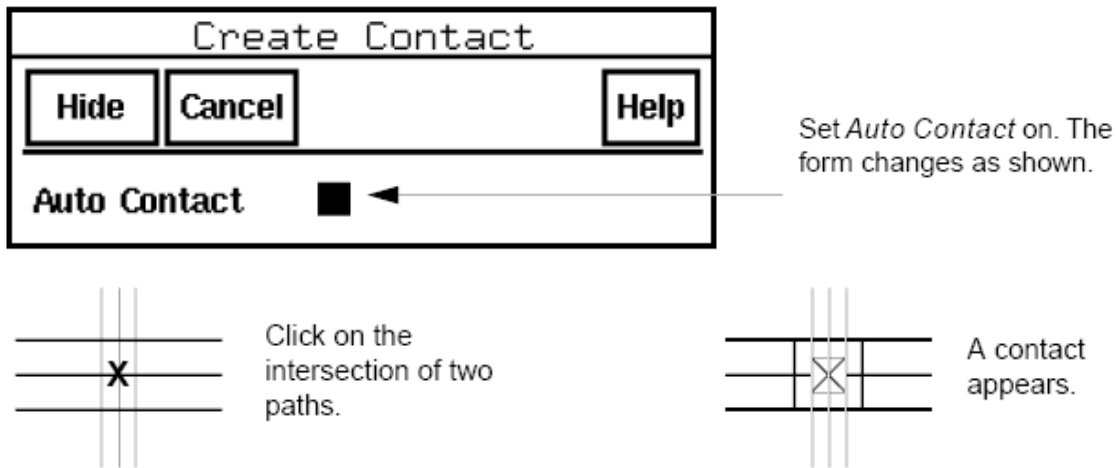


The resulting pin array



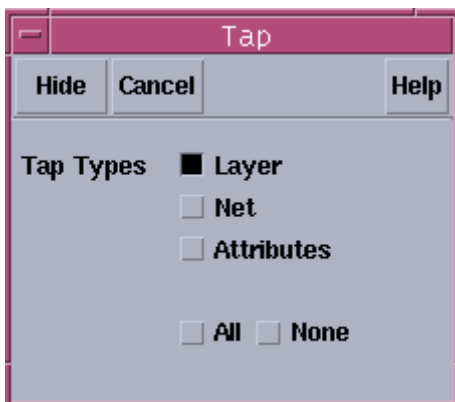
## 5.在已存在的两个path交错的地方自动打孔

两层之间的互连，可以手动打孔，也可以用**Create Contact** 自动打孔，但是必须在**两条path交错**的地方，而且两层之间有孔的定义（见Path stitching中contact的定义），不能跨层打孔（如M1—M3之间），不能在斜线上自动打孔，两条path不受cell的层次的限制，只要是可见的。如图，选中**Auto Contact**，通孔中心会自动移到两条path的中心线交点处，但要注意**交点是否在格点上**



## 6. 关于Tap的使用

Tap是用来快速捕捉任意一个图形的相关属性，捕捉到的信息会自动赋给将要生成的Object上包括Layer信息，Net的名字，其它属性（如path的width，endtype等等）



### a. 用Tap来快速选择entry Layer

选择 *Edit - Tap* [t]，默认的是选择的Layer是边界最接近鼠标指针（或者边界与指针重合）的图形的Layer，如果两层完全重合，则有一层可能一直也选不中。

因此可以设置环境变量 `layerTapCycle` 为 `t` or `nil`，可以改变捕捉的方法  
在CIW敲入命令（只对这个icfb窗口适用）：

```
envSetVal("layout" "layerTapCycle" "boolean t")
```

或者在根目录下的.cdsenv文件中加入一行，每次启动icfb就会调用这个环境变量  
`layout layerTapCycle boolean t`

设成 `nil` 时（默认值）选择的Layer是边界最接近鼠标指针（或者边界与指针重合）的图形的Layer

设成 `t` 时，在重复的图形上连续点击，可以依次轮流的选择不同的Layer，直到选到你想要的为止。

b. 用Tap捕捉图形信息，并传给将要生成的图形上

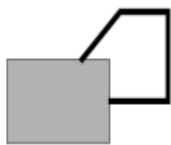
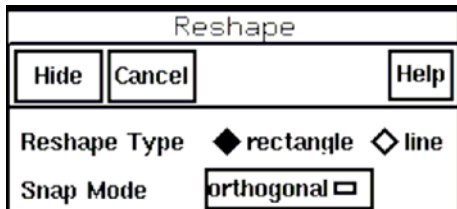
将Tap Types设为All（net名不一样的话去掉Net，不过我们很少设置net name），点击图形后会捕捉到它的Layer信息，Net的名字，其它属性等，如下图所示的示例，不同的layer，width，endtype，Tap后再画path，path里的设置都和你tap的图形一样了



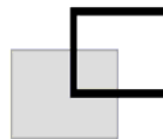
## 7. Reshape Objects

**Reshape**顾名思义就是改变图形的形状，但要注意几点：

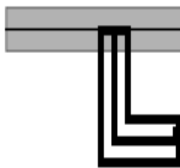
1. **Reshape Type**设为**rectangle**时，**rectangle**的一角必须与原来的图形交错
2. **Reshape Type**设为**line**时，第一点和最后一点必须在原来图形的边线上，另外**Reshape path**只能选择**line**，第一点必须在原来**path**的中线上
3. **Reshape**的图形在完成之前会高亮显示，点击**右键**会**切换不同的高亮区域**，确认是否是最终的图形形状



The first and last points must touch the original shape.

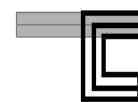
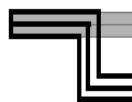


One corner of the rectangle must intersect the original shape.



Start the new segment from the path centerline.

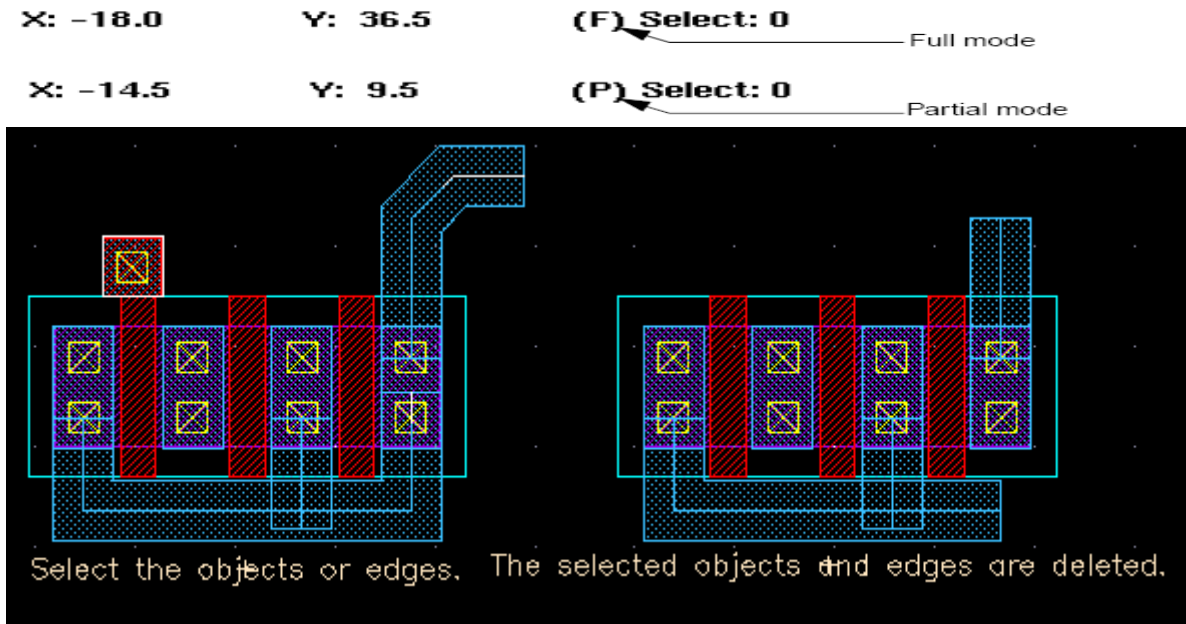
Click right to toggle between highlighting the reshaped path options.



Reshape options

## 8. 关于部分选择及相关的操作

在stretch时可以选择一部分操作，但通常其他情况下只能选择全部，因为默认是full select mode，stretch是自动切换到partial select mode时，其实按F4键就可以切换这两种模式，这种模式下可以stretch，delete等



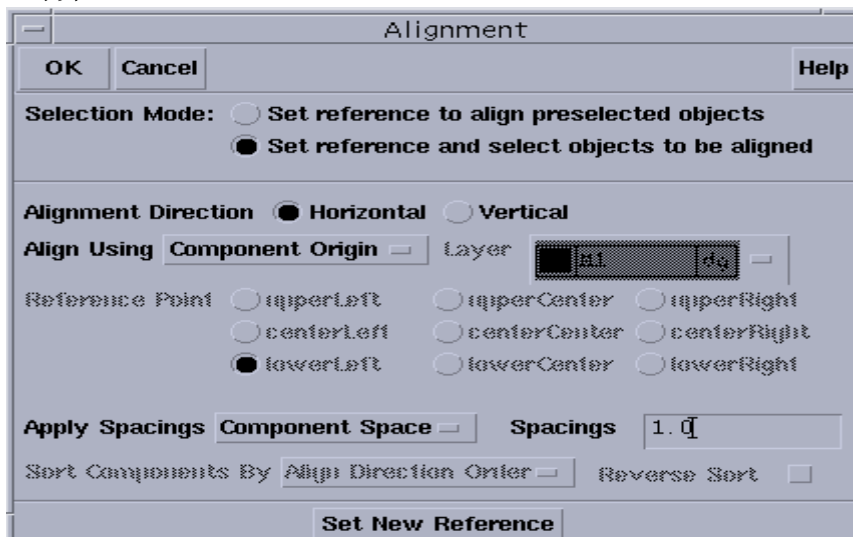
## 9. 关于图形的对齐

我们经常会有一些图形需要对齐，例如一组横向摆放的pad，要求对齐且间隔相等，手动对齐比较麻烦，在Edit->Other->Align有对齐的命令，可以用来对齐两个或多个图形。对齐图形有 preselect和postselect 两种方式。

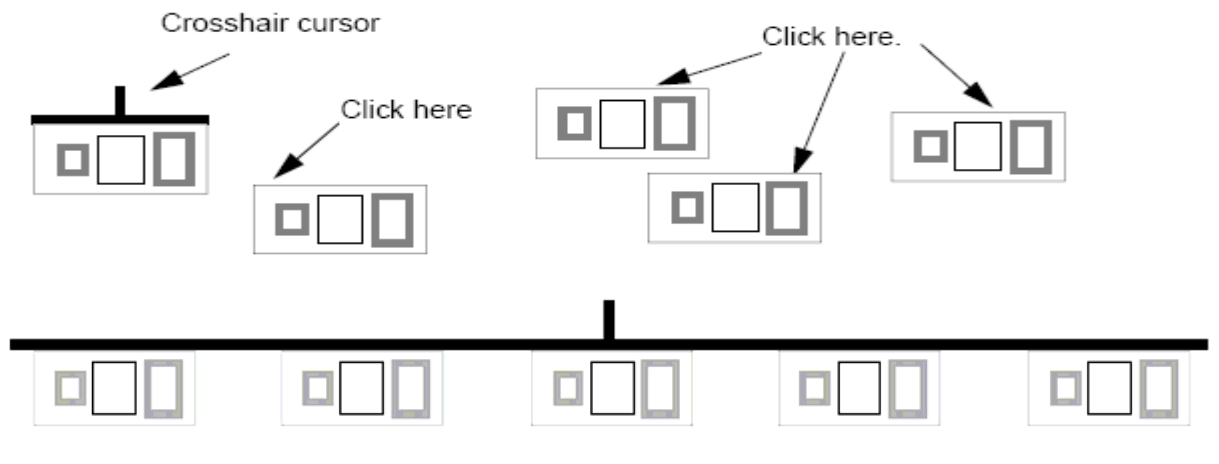
### a. Postselect方式

把Selection Mod 设置为 *Set reference and select objects to be aligned*.

点击Set New Reference，先选择一个图形的边界作为参考边，然后选择其他图形与这个参考边对齐。







### b. Preselect方式

设置 *Selection Mod* 为 *Set reference to align preselected objects*.

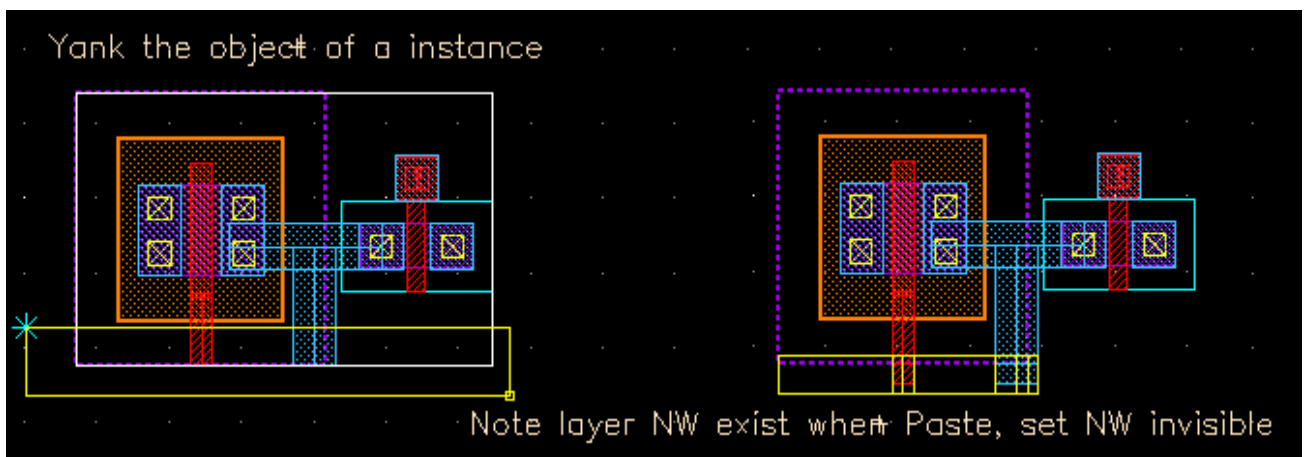
先选择未对齐的图形，然后点击 *Set New Reference and* 在 *layout* 窗口选择一个对齐点，所有选择的图形就会对齐。

当然对齐还有其他的一些设置，但要注意这里的对齐操作是对一个整体而言(如一个 *instance*, *polygon* 等)，如果几个图形需要组合在一起，参与与其他单元的对齐，先将这几个图形 *make cell* 或者用 *attach* (而且被 *attach* 的图形不能被选中)，否则这几个图形将被拆开。

## 10. Yanking & Pasting图形 (即复制-粘贴)

我们经常用 *copy* 命令来复制图形，但复制的都是一个整体 (如一个 *instance*, 一条 *path* 等)，*Yank* 命令可以复制 (相当于切割) 一部分图形到一个临时的 *buffer*，而 *Paste* 命令把 *buffer* 中的图形复制到 *layout* 窗口中。

*Yank & Paste* 与 *Copy* 的区别在于 *Yank* 的图形是 *chop object* 而来，*Copy* 的图形是 *Full select object* 而来，*Yank/Paste* 可以看作是“少量信息切割”的 *copy*。需要注意的是 *Yank* 图形层次是可以控制的，部分选择的图形将被 *chop*，*Paste* 的图形是打平的 (注意 *paste* 的图形的 *layer*，如果有不需要的 *layer*，设为 *invisible* 可以不复制这些层)；全部选择的图形依然保存层次化的结构。如图是 *Yank/Paste* 下个层次的图形的例子：

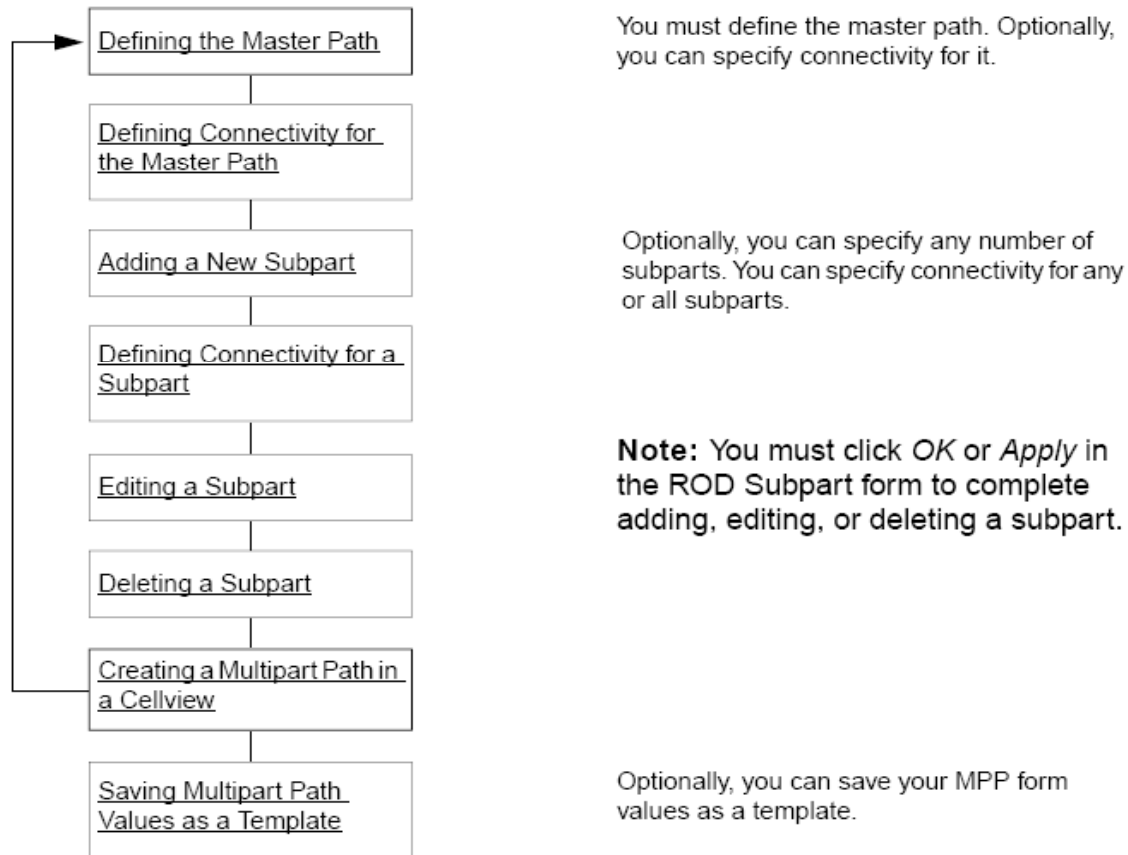


## 11.生成Multipart Paths

*Multipart path*是指一条path包含有多部分 ( $\geq 2$ ) 与其相关的图形, *Create Multipart Path*命令是用来生成一条path并包含有其他相关的图形 (relative object design ---ROD) 信息, 你可以用它来画一条简单的path, 也可以一条有几部分组成的复杂*multipart path*, 例如: **guard ring, transistor, bus**, 或者 **shielded path**。

*Multipart path*同样可以进行stretch, chop, reshape等操作, 并且对应的ROD图形会根据master path的变化自动重新画。

### a. 画Multipart Paths的流程, 如下图

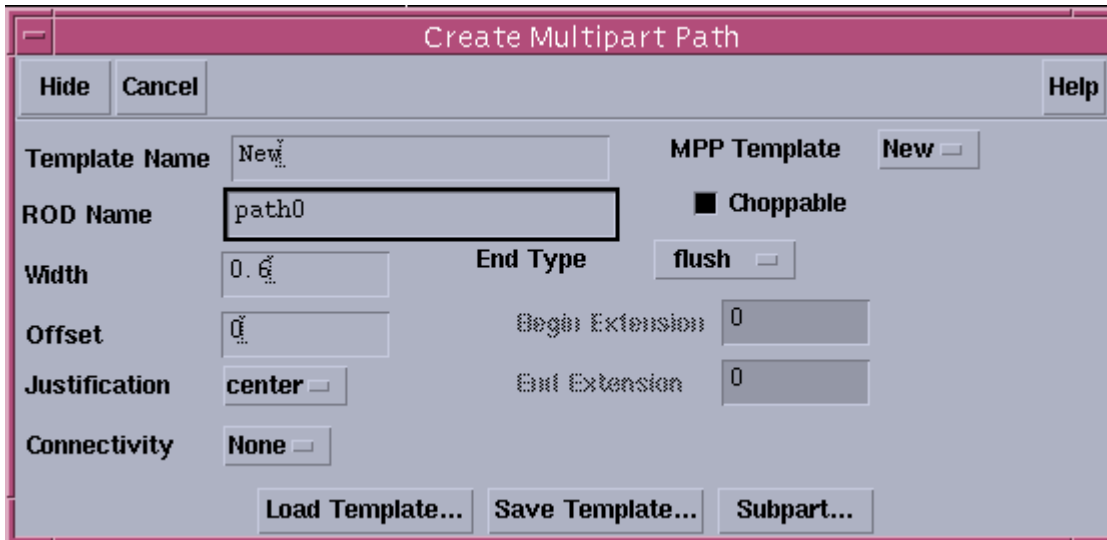


### b. 定义Master Path

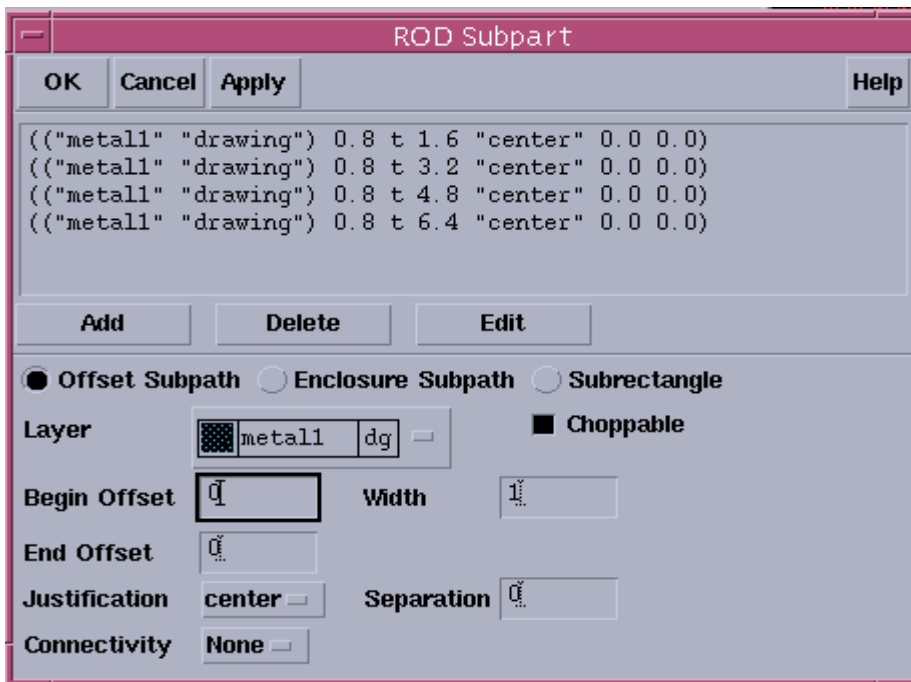
对于master path的*Choppable*选项:

使**master path**和**所有的subparts**都**choppable**, 打开*Choppable*选项;

使master path不能choppable (但是**subparts**既可以**choppable**也可以不**choppable**), 关闭*Choppable*.



c. 添加Subpart, 可以添加path, rectangle等



d. Subpath和Master Path的位置关系见下表

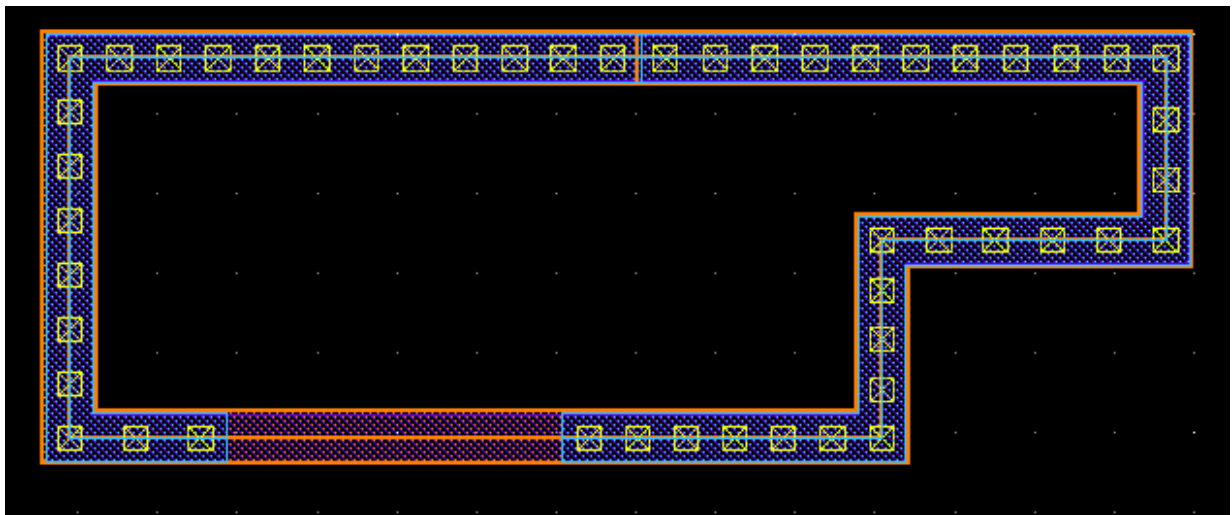
Separation	Center Justification	Left Justification	Right Justification
Zero	Subpath centerline on master path centerline	Left edge of master path coincident with right edge of subpath	Right edge of master path coincident with left edge of subpath

Positive number	Subpath centerline on left side of master path centerline	Left edge of master path on right side of right edge of subpath	Right edge of master path on left side of left edge of subpath
Negative number	Subpath centerline on right side of master path centerline	Left edge of master path on left side of right edge of subpath	Right edge of master path on right side of left edge of subpath

Table 10-3 Position of Subrectangles in Relation to Master Path			
Separation	Center Justification	Left Justification	Right Justification
Zero	Center of width of subrectangles on master path centerline	Left edge of master path coincident with right edge of subrectangles	Right edge of master path coincident with left edge of subrectangles
Positive number	Center of width of subrectangles on left side of master path centerline	Left edge of master path on right side of right edge of subrectangles	Right edge of master path on left side of left edge of subrectangles
Negative number	Center of width of subrectangles on right side of master path centerline	Left edge of master path on left side of right edge of subrectangles	Right edge of master path on right side of left edge of subrectangles

e. 实例: 用 *Multipart path* 画 Guard Ring

其中 master path 选择有源区 Active 层, 而 metal1 和注入层 (如 PPlus) 作为 enclosure subpath, 而接触孔 CT 设为 subrectangles, metal1 和 CT 孔设为 choppable, 具体方法这里不在详述, 结果如下图, 后面附有用 *Multipart path* 生成 Bus 和 Tap 的 skill 程序。

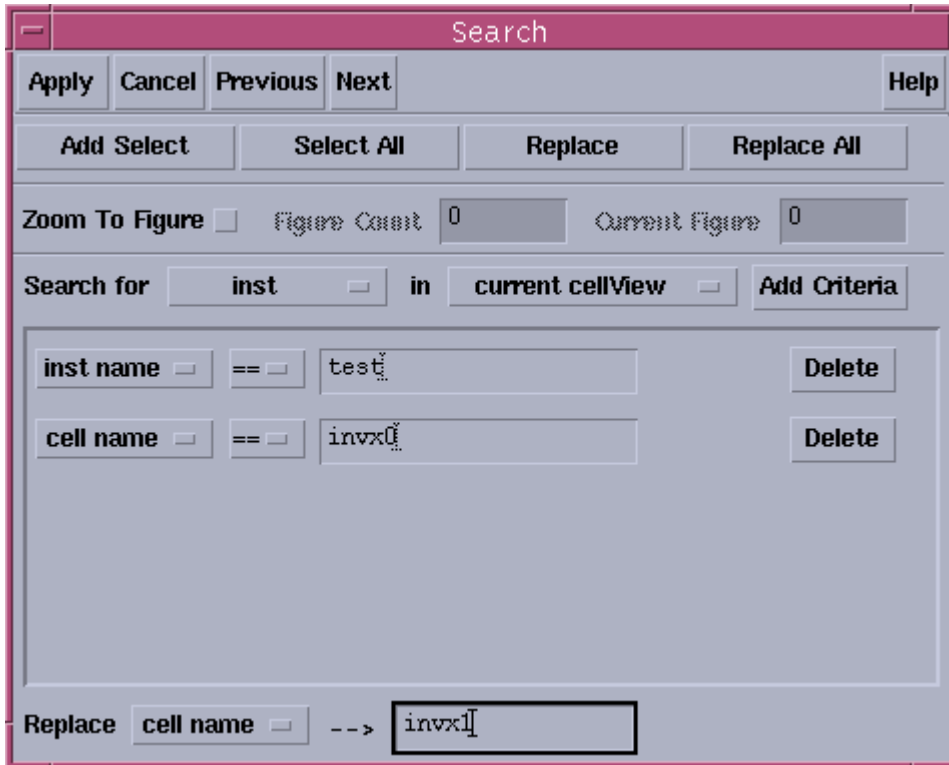


## 12. Search and replace的应用

查找和替换是个很有用的功能，可以快速查找一个图形对象，以及其属性，并且可以批量替换。如图，你可以添加多个搜索条件，

在用 == 和 != 时，可以用通配符：

- \* 可以替代任意数的任意字符
- . 可以替代当前位置的任一字符
- [] 可以替代一个范围的字符



### 13. 提高软件速度的一些环境变量的优化

Environment Variable	Old	New	Effect
dotsOn	t	nil	Dots are not displayed.
displayStretchHandles	t	nil	Stretch handles are not displayed.
filterSize	3	6	Objects with one side of less than 6 pixels are not displayed. Instances not at the stop level with two sides of less than 6 pixels are displayed according to the filterSizeDrawingStyle setting.
filterSizeDrawingStyle	outlined	empty	Now used only for instances which are not at the display stop level. When set to empty, instances are not displayed.
gravityOn	t	nil	Gravity is turned off.
openToStopLevel	nil	t	All data is loaded prior to rendering.
originMarkersOn	t	nil	Instance origin markers are not displayed.
useTrueBBox	t	nil	Displays only the instance master bounding box.

14. 这里的skill程序是用来快速定义 *Multipart path* 的 template, 方便一些参数的修改, 下面是用 *Multipart path* 生成 Bus 的 skill 程序

```
procedure(DefineBusMPPTemplate(BusNumber BusWidth BusSpace)
let((CV BusLayer Tech SubListN SubPathList)
  CV = geGetEditRep()
  Tech = techGetTechFile(CV)
  if(Tech==nil then
    error("Can't read the techfile!")
  )
)
```

```

BusLayer = leGetEntryLayer()
while(BusNumber>1
  --BusNumber
  SubListN = list(
    ?layer BusLayer
    ?width BusWidth
    ?choppable t
    ?sep BusNumber*(BusWidth+BusSpace)
    ?justification "center"
    ?beginOffset 0.000
    ?endOffset 0.000
  )
  SubPathList = cons( SubListN SubPathList )
);while

```

**;The template name is : BusMPP**

```

leDefineMPPTemplate(
  ?techId Tech
  ?name "BusMPP"
  ?layer BusLayer
  ?width BusWidth
  ?choppable t
  ?endType "flush"
  ?beginExt 0.000
  ?endExt 0.000
  ?justification "center"
  ?offset 0.000
  ?offsetSubPath SubPathList
);
);let
);procedure

```

**/\*\*\*\*\*\*Top procedure to create form\*\*\*\*\*\*/**

**procedure( CreateBus()**

```

let((BusWidth BusSpace BusNumber BusNumberField BusWidthField BusSpaceField
  MyLabelField )

```

**;;; create int field for BusNumber value.**

```

BusNumberField = hiCreateIntField(
  ?name 'BusNumberField
  ?prompt "Bus Number:"
  ?range '(1 1000)
  ;?value 3
  ?defValue 5
  ;?callback "println( hiGetCurrentForm()->BusNumberField-> value )"
)

```

**;;; create float field for BusWidth & BusSpace value.**

```

BusWidthField = hiCreateFloatField(
  ?name 'BusWidthField
  ?prompt "Bus Width(um):"
  ?range '(0.001 1000.0)
  ;?value 0.6
  ?defValue 0.6
  ;?callback "println( hiGetCurrentForm()->BusWidthField-> value )"
)

```

```

BusSpaceField = hiCreateFloatField(
    ?name      'BusSpaceField
    ?prompt    "Bus Space(um):"
    ?range     '(0.001 1000.0)
    ;?value    0.6
    ?defValue  0.6
    ;?callback "println( hiGetCurrentForm()->BusSpaceField-> value )"
)

```

;;; create the label field

```

MyLabelField = hiCreateLabel(
    ?name      'MyLabelField
    ?labelText "Make sure select the proper Layer first!!!" ;"Create by wanyb"
    ?justification 'right
    ?enabled   t
    ;?font     "roman"
    ;?invisible t
)

```

;;; create the form

```

hiCreateAppForm(
    ?name 'CreateBusForm
    ?formTitle "Create Bus with Multipart Path"
    ?callback "DefineBusMPPTemplate(hiGetCurrentForm()->BusNumberField->value
                                     hiGetCurrentForm()->BusWidthField->value
                                     hiGetCurrentForm()->BusSpaceField->value)"

    ?fields
    list(
        list(BusNumberField 20:5      180:30 120 )
        list(BusWidthField  20:25     180:30 120 )
        list(BusSpaceField  20:45     180:20 120 )
        list(MyLabelField   40:80      180:25  )
    )
    ?unmapAfterCB t
    ;?formType    'options
    ;?buttonLayout 'OKCancelDefLast
    ;?help        'g_help
)

```

```
); hiCreateAppForm
```

```
;hiDisplayForm( CreateBusForm )
```

```
);let
```

```
);procedure
```

;the current value displayed in the form is the last value you input.

```

procedure(DisplayCreateBusForm()
    if((boundp('CreateBusForm) == nil)
        CreateBus()
    )
    hiDisplayForm( CreateBusForm )
)

```

```
hiSetBindKey("Layout" "Ctrl<Key>m" "{DisplayCreateBusForm() leHiCreateMPP()}")
```



## 15.用 *Multipart path* 生成 Tap 的 skill 程序

```
procedure(DefineTapMPPTemplate())
```

```
let((EncSubPathL SubRectList EncSubPath_M1 EncSubPath_Imp EncSubPath_NW CT_N  
SubRectListN)
```

```
;define encSubPath list
```

```
EncSubPath_M1 = list(  
    ?layer      M1LPP  
    ?enclosure  AAEncCT-M1EncCT  
    ?choppable  ChoppableM1  
    ?beginOffset M1BeginOffset  
    ?endOffset  M1EndOffset  
)
```

```
EncSubPath_Imp = list(  
    ?layer      ImpLPP  
    ?enclosure  -1*ImpEncAA  
    ?choppable  ChoppableImp  
    ?beginOffset ImpBeginOffset  
    ?endOffset  ImpEndOffset  
)
```

```
EncSubPathL = cons( EncSubPath_M1 EncSubPathL )
```

```
EncSubPathL = cons( EncSubPath_Imp EncSubPathL )
```

```
when(TapType == "NTap"
```

```
    EncSubPath_NW = list(  
        ?layer      NWLPP  
        ?enclosure  -1*NWEncAA  
        ?choppable  ChoppableNW  
        ?beginOffset NWBeginOffset  
        ?endOffset  NWEndOffset  
    )
```

```
        EncSubPathL = cons( EncSubPath_NW EncSubPathL )
```

```
);when
```

```
;define subRect list
```

```
CT_N = CT_Num
```

```
while(CT_N > 0
```

```
    SubRectListN = list(  
        ?layer      CTLPP  
        ?width      CTWidth  
        ?length     CTWidth  
        ?choppable  ChoppableCT  
        ?sep        -1*(AAEncCT+CT_N*CTWidth+(CT_N-1)*CTSpace)  
        ?justification "left"  
        ?space      CTSpace  
        ?beginOffset CTBeginOffset  
        ?endOffset  CTEndOffset  
        ?beginSegOffset 0.000  
        ?endSegOffset 0.000  
        ?gap        CTGapType  
    )
```

```
    SubRectList = cons( SubRectListN SubRectList )
```

```
    CT_N--
```

```
);end while
```

```
;The template name is : NTap/PTap
```

```
leDefineMPPTemplate(  
  ?techId      Tech  
  ?name        TapName  
  ?layer       AALPP  
  ?width       CT_Num*CTWidth+(CT_Num-1)*CTSpace+2*AAEncCT  
  ?choppable   ChoppableAA  
  ?endType     "flush"  
  ?beginExt    0.000  
  ?endExt      0.000  
  ?justification "center"  
  ?offset      0.000  
  ?encSubPath  EncSubPathL  
  ?subRect     SubRectList  
); end leDefineMPPTemplate
```

```
);end let
```

```
);end procedure
```

```
;;;form & field callback procedure
```

```
procedure( CreateTapFormCB()
```

```
let((TapName CTLPP AALPP M1LPP ImpLPP NWLPP CTWidth CTSpace AAEncCT ImpEncAA  
NWEncAA CTGapType
```

```
  Choppable ChoppableAA ChoppableImp ChoppableNW ChoppableM1 ChoppableCT
```

```
M1BeginOffset
```

```
  M1EndOffset ImpBeginOffset ImpEndOffset NWBeginOffset NWEndOffset CTBeginOffset
```

```
CTEndOffset
```

```
)
```

```
TapName = CreateTapForm->TapNameField->value
```

```
CTLPP = parseString( car(last(CreateTapForm->CTLPPField->value)) " ()" )
```

```
AALPP = parseString( car(last(CreateTapForm->AALPPField->value)) " ()" )
```

```
M1LPP = parseString( car(last(CreateTapForm->M1LPPField->value)) " ()" )
```

```
ImpLPP = parseString( car(last(CreateTapForm->ImpLPPField->value)) " ()" )
```

```
;NWLPP = parseString( car(last(CreateTapForm->NWLPPField->value)) " ()" )
```

```
CT_Num = CreateTapForm->CT_NumField->value
```

```
CTWidth = CreateTapForm->CTWidthField->value
```

```
CTSpace = CreateTapForm->CTSpaceField->value
```

```
AAEncCT = CreateTapForm->AAEncCTField->value
```

```
M1EncCT = CreateTapForm->M1EncCTField->value
```

```
ImpEncAA = CreateTapForm->ImpEncAAField->value
```

```
;NWEncAA = CreateTapForm->NWEncAAField->value
```

```
CTGapType = CreateTapForm->CTGapTypeField->value
```

```
case( CreateTapForm->ChoppableField->value
```

```
  ("M1&CT choppable"
```

```
    ChoppableAA=ChoppableImp=ChoppableNW=nil
```

```
    ChoppableM1=ChoppableCT=t )
```

```
  ("All choppable"
```

```
    ChoppableAA=ChoppableImp=ChoppableNW=ChoppableM1=ChoppableCT=t )
```

```
  ("None choppable"
```

```
    ChoppableAA=ChoppableImp=ChoppableNW=ChoppableM1=ChoppableCT=nil )
```

```
);case
```

```

;;;a>=b ? a : b, used for chop Tap DRC check when M1EncCT > AAEncCT
CTBeginOffset = CTEndOffset =
  -1*((M1EncCT>=AAEncCT&&M1EncCT)||M1EncCT<AAEncCT&&AAEncCT))

if( CreateTapForm->RingPField->value == nil then
  M1BeginOffset = M1EndOffset = M1EncCT-AAEncCT
  ImpBeginOffset = ImpEndOffset = ImpEncAA
  when(TapType=="NTap"
    NWLPP = parseString( car(last(CreateTapForm->NWLPPField->value)) " ()" )
    NWEncAA = CreateTapForm->NWEncAAField->value
    NWBeginOffset = NWEEndOffset = NWEncAA
  );when
else
  M1BeginOffset = M1EncCT-AAEncCT  M1EndOffset = AAEncCT-M1EncCT
  ImpBeginOffset = ImpEncAA  ImpEndOffset = -1*ImpEncAA
  ;when(-CTBeginOffset<CTSpace/2  CTBeginOffset = CTEndOffset = -CTSpace/2 )
  ;used for CT space DRC check when AAEncCT + (-CTEndOffset) < CTSpace
  when(-CTEndOffset<CTSpace-AAEncCT  CTEndOffset = AAEncCT-CTSpace )
  when(TapType=="NTap"
    NWLPP = parseString( car(last(CreateTapForm->NWLPPField->value)) " ()" )
    NWEncAA = CreateTapForm->NWEncAAField->value
    NWBeginOffset = NWEncAA  NWEEndOffset = -1*NWEncAA
  );when
);if

DefineTapMPPTemplate()
);let
);procedure

procedure(TapTypeFieldCB)
  TapType = CreateTapForm->TapTypeField->value
  CT_Num = CreateTapForm->CT_NumField->value
  if(CreateTapForm->RingPField->value == t then
    CreateTapForm->TapNameField->value = get_pname( concat(TapType "_" CT_Num "CT"
  "_Ring" ) )
  else
    CreateTapForm->TapNameField->value = get_pname( concat(TapType "_" CT_Num "CT" ) )
  );if
  if(TapType == "PTap" then
    hiDeleteFields(CreateTapForm list('NWLPPField 'NWEncAAField) )
  else
    hiAddFields(CreateTapForm
      list(list( NWLPPField 460:85 160:30 75 )
        list( NWEncAAField 460:180 210:25 150 )
      )
    )
  );if
);end procedure --- TapTypeFieldCB

procedure(RingPFieldCB)
  TapType = CreateTapForm->TapTypeField->value
  CT_Num = CreateTapForm->CT_NumField->value
  if( CreateTapForm->RingPField->value == nil then
    CreateTapForm->TapNameField->value = get_pname( concat(TapType "_" CT_Num "CT" ) )
  else
    CreateTapForm->TapNameField->value = get_pname( concat(TapType "_" CT_Num "CT"
  "_Ring" ) )
  );if

```

```
);end RingPFieldCB
```

```
/****** procedure to create Guard ring form******/
```

```
procedure( CreateTap()
```

```
let((CT_Num InitLayerList AllLayerList SelectLayerList  
    TapTypeField TapNameField CTLPPField AALPPField M1LPPField ImpLPPField  
    CT_NumField CTWidthField CTSpaceField AAEncCTField M1EncCTField ImpEncAAField  
    ChoppableField CTGapTypeField RingPField MyLabelField ;NWEncAAField NWLPPField  
    )
```

```
;init the SelectLayerList list or all layer in the tech file will be displayed.
```

```
InitLayerList = list(  
    list("M1" "drawing")  
    list("ACT" "drawing") list("AA" "drawing")  
    list("CT" "drawing") list("CON" "drawing")  
    list("PP" "drawing") list("SP" "drawing")  
    list("NP" "drawing") list("SN" "drawing")  
    list("NW" "drawing")  
    )  
AllLayerList = leGetValidLayerList(Tech)  
SelectLayerList = setof(ALayer InitLayerList  
    member(ALayer AllLayerList)  
    )  
when( length(SelectLayerList) != 6 SelectLayerList = AllLayerList )
```

```
::: create Radio field for TapType value.
```

```
TapTypeField = hiCreateRadioField(  
    ?name 'TapTypeField  
    ?prompt "Select the Tap Type:"  
    ?choiceslist( "NTap" "PTap" )  
    ?value "NTap"  
    ?defValue "NTap"  
    ?callback '( "TapTypeFieldCB()" )  
    )
```

```
::: create String field for TapName value.
```

```
TapNameField = hiCreateStringField(  
    ?name 'TapNameField  
    ?prompt "Tap template name:"  
    ?value "NTap"  
    ?defValue "PTap"  
    ;?callback "TapName = CreateTapForm->TapNameField->value "  
    ?editable t  
    )
```

```
::: create Layer cyclic field for CTLPP, AALPP, M1LPP, ImpLPP, NWLPP value.
```

```
CTLPPField = hiCreateLayerCyclicField(  
    Tech  
    "CT layer:"  
    ;t_callback  
    "CTLPP=parseString( car(last(CreateTapForm->CTLPPField->value)) \" ()\" ) "  
    SelectLayerList  
    nil  
    'CTLPPField
```

```

)
AALPPField = hiCreateLayerCyclicField(
    Tech
    "AA layer:"
    "AALPP=parseString( car(last(CreateTapForm->AALPPField->value)) \" ()\" ) "
    SelectListLayerList
    nil
    'AALPPField
)
M1LPPField = hiCreateLayerCyclicField(
    Tech
    "M1 layer:"
    "M1LPP=parseString( car(last(CreateTapForm->M1LPPField->value)) \" ()\" ) "
    SelectListLayerList
    nil
    'M1LPPField
)
ImpLPPField = hiCreateLayerCyclicField(
    Tech
    "Imp layer:"
    "ImpLPP=parseString( car(last(CreateTapForm->ImpLPPField->value)) \" ()\" ) "
    SelectListLayerList
    nil
    'ImpLPPField
)
NWLPPField = hiCreateLayerCyclicField(
    Tech
    "NW layer:"
    "NWLPP=parseString( car(last(CreateTapForm->NWLPPField->value)) \" ()\" ) "
    SelectListLayerList
    nil
    'NWLPPField
)

```

;;; create int field for tap width--CT\_Num value.

```

CT_NumField = hiCreateIntField(
    ?name      'CT_NumField
    ?prompt    "CT Number for Tap Width:"
    ?range     '(1 1000)
    ?value     1
    ?defValue  2
    ?callback  "CT_Num = CreateTapForm->CT_NumField->value "
)

```

;;; create float field for CTWidth, CTSpace, AAEncCT, M1EncCT, ImpEncAA, NWEncAA value.

```

CTWidthField = hiCreateFloatField(
    ?name      'CTWidthField
    ?prompt    "CT Width(um):"
    ?range     '(0.001 1000.0)
    ?value     0.4
    ?defValue  0.3
    ;?callback "CTWidth = CreateTapForm->CTWidthField->value "
)
CTSpaceField = hiCreateFloatField(
    ?name      'CTSpaceField
    ?prompt    "CT Space(um):"

```

```

        ?range      '(0.001 1000.0)
        ?value      0.4
        ?defValue   0.3
        ;?callback  "CTSpace = CreateTapForm->CTSpaceField->value "
    )
    AAEncCTField = hiCreateFloatField(
        ?name      'AAEncCTField
        ?prompt    "AA Enclosure CT (um):"
        ?range     '(0.001 1000.0)
        ?value     0.15
        ?defValue  0.15
        ;?callback "AAEncCT = CreateTapForm->AAEncCTField->value "
    )
    M1EncCTField = hiCreateFloatField(
        ?name      'M1EncCTField
        ?prompt    "M1 Enclosure CT (um):"
        ?range     '(0.001 1000.0)
        ?value     0.2
        ?defValue  0.15
        ;?callback "M1EncCT = CreateTapForm->M1EncCTField->value "
    )
    ImpEncAAField = hiCreateFloatField(
        ?name      'ImpEncAAField
        ?prompt    "Imp Enclosure AA (um):"
        ?range     '(0.001 1000.0)
        ?value     0.25
        ?defValue  0.04
        ;?callback "ImpEncAA = CreateTapForm->ImpEncAAField->value "
    )
    NWEncAAField = hiCreateFloatField(
        ?name      'NWEncAAField
        ?prompt    "NW Enclosure AA (um):"
        ?range     '(0.001 1000.0)
        ?value     0.2
        ?defValue  0.15
        ;?callback "NWEncAA = CreateTapForm->NWEncAAField->value "
    )

```

;;; create Cyclic field for Choppable, CTGapType value.

```

    ChoppableField = hiCreateCyclicField(
        ?name      'ChoppableField
        ?prompt    "Choppable: "
        ?choices   list("M1&CT choppable" "All choppable" "None choppable")
        ?value     "M1&CT choppable"
        ?defValue  "All choppable"
        ;?callback "ChoppableFieldCB() "
    )
    CTGapTypeField = hiCreateCyclicField(
        ?name      'CTGapTypeField
        ?prompt    "CT rect GapType: "
        ?choices   list("distribute" "minimum" )
        ?value     "distribute"
        ?defValue  "minimum"
        ;?callback "CTGapType = CreateTapForm->CTGapTypeField->value "
    )

```

;;; create Boolean field for Ring Tap subpath offset .

```
RingPField = hiCreateBooleanButton(  
  ?name 'RingPField  
  ?buttonText "Tap is a Ring? "  
  ?value nil  
  ?defValue t  
  ?buttonLocation 'right  
  ?callback "RingPFieldCB()"  
)
```

;;; create the label field

```
MyLabelField = hiCreateLabel(  
  ?name      'MyLabelField  
  ?labelText "Create by wanyb"  
  ?justification 'right  
  ?enabled   nil  
  ;?font     "roman"  
  ;?invisible t  
)  
LayerSelectLabelField = hiCreateLabel(  
  ?name      'LayerSelectLabelField  
  ?labelText "Select the proper layer: "  
  ?justification 'left  
  ?enabled   nil  
  ;?invisible t  
)  
DesingRuleLabelField = hiCreateLabel(  
  ?name      'DesingRuleLabelField  
  ?labelText "Define following rules: "  
  ?justification 'left  
  ?enabled   nil  
  ;?invisible t  
)
```

;;; create the form

```
CreateTapForm = hiCreateAppForm(  
  ?name      'CreateTapForm  
  ?formTitle "Create Tap with Multipart Path"  
  ?callback  "CreateTapFormCB()"  
  ?fields    list(  
    list( TapTypeField  20:5360:25  160 )  
    list( TapNameField  20:30  300:25  140 )  
  
    list( LayerSelectLabelField  20:65  360:20 )  
    list( M1LPPField 20:85  160:30  75 )  
    list( AALPPField 240:85  160:30  75 )  
    list( NWLPPField  460:85  160:30  75 )  
    list( CTLPPField 20:120  160:30  75 )  
    list( ImpLPPField  240:120 160:30  75 )  
  
    list( DesingRuleLabelField  20:165  360:20 )  
    list( CTWidthField  20:180  160:25  100 )  
    list( CTSpaceField  20:210  160:25  100 )  
    list( CT_NumField   20:240  210:25  160 )  
    list( AAEncCTField  240:180 210:25  150 )
```

```

list( M1EncCTField  240:210 210:25  150 )
list( ImpEncAAField 240:240 210:25  150 )
list( NWEncAAField  460:180 210:25  150 )

list( ChoppableField 20:280  210:25  75 )
list( CTGapTypeField240:280 220:25  120 )
list( RingPField  470:280 120:25  90 )

list( MyLabelField  460:320 220:20 )

)

;?buttonLayout 'OKCancelDefLast
?unmapAfterCB t
;?formType? 'options
;?help "hehehe"

); hiCreateAppForm
;hiDisplayForm( CreateTapForm )

);let
);procedure

;the current value displayed in the form is the last value you input.
procedure(DisplayCreateTapForm()
  if(boundp('CreateTapForm)==nil || CreateTapForm==nil then
    CreateTap()
  )
  hiDisplayForm(CreateTapForm)
)

;;; init the techfile then set the binkkey.
Tech = techGetTechFile(geGetEditRep())
if(boundp('Tech)==nil || Tech==nil then
  error("Can't read the techfile! Open the Layout First!")
else
  hiSetBindKey("Layout" "Ctrl<Key>t" "{DisplayCreateTapForm() leHiCreateMPP()}")
);if

```