Expert Parametric Wires

Introduction

Expert Parametric Wire (Pwire) is a complex group of objects containing, a single wire named master wire, any number of subparts such as enclosure wires, offset wires and sets of rectangles. Pwire objects enable extremely quick and efficient creation and editing of guard rings and shielded paths which are increasingly important due to higher integration density of IC designs.

The number of subparts for each type, its properties and placement with respect to the master wire is defined by a set of pwire parameters. Pwire parameters for the master wire and each subpart are defined in the 'Parametric Wire' Numeric Input Form. Once the parametric values are entered you can create a parametric wire in the ordinary way by point-and-click mouse operation to enter points in the current cell window. When you create a Pwire group the last one is treated as a single object in many edit operations such as modify, stretch, copy, paste, move, duplicate, etc.

You can store a particular set of Pwire parameters into the *Expert* technology (internal binary *.eld or *.sld project file or external ASCII *.tcn file) as a template. The Pwire template is a predefined set of pwire parameters that define a particular Pwire, for example, a guard ring. The Pwire template can be chosen in 'Parametric Wire' Numeric Input Form (see Figure 1). Once you select the Pwire template each field in the Numeric Input form will be filled with the matching parameter value stored in the template. You can make changes for some parameters values before entering points for Pwire in a cell window.

Creating Pwires

You can create a Pwire object using the *Expert* menu command Edit>>Create Object>>Parametric Wire or Xi command PWIRE.

Figure 1. Pwire Creation Numeric Input Panel

Edit>>Create Object>>Parametric Wire Command

Once you select the command, the numeric input dialog similar to the ordinary wire numeric input panel pops up (see Figure 1). But there are a couple of additional fields and buttons.

SILVACO INTERNATIONAL

Figure 2. Subparts Dialog Panel

Subparts dialog panel allows add, remove, and modify subparts for the current Pwire.

Pwire object can include any number of subparts such as enclosure wires, offset wires and sets of rectangles. Pwire subparts list is shown on the top part of the Subparts dialog panel. When you select existing subpart then values of its parameters will be automatically filled in the corresponding fields. You can edit the values of parameters and click 'Update' button to apply changes to the selected subpart. 'Delete' button is to delete the highlighted subpart.
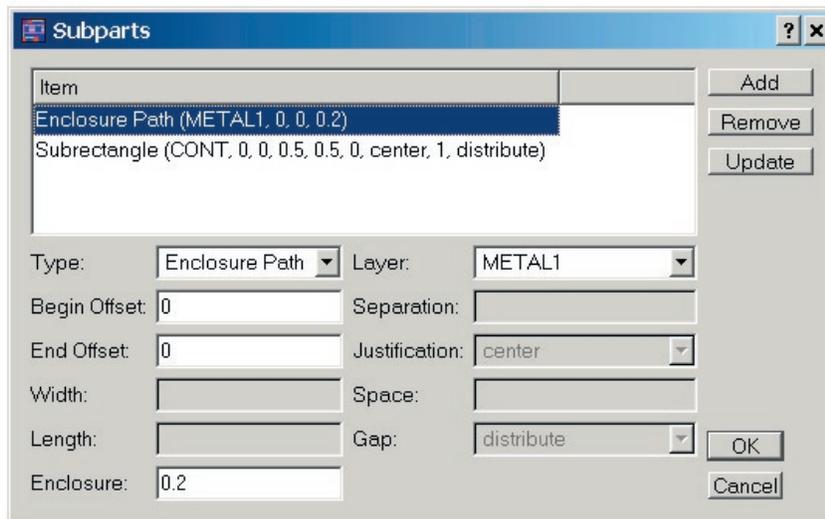
To add a new subpart first of all you

Template allows selecting template stored in the technology with predefined pwire parameters.

Name is to assign a unique name for the created pwire object. If the name is not specified then a system defined unique name will be assigned.

Subparts… button opens 'Subparts' dialog panel to allow you to add, remove, or update subparts of Pwire object (see Figure 2).

Input fields Layer, Width, End, Joint defines attributes of the master wire. Vertices for master wire can be added by a mouse click or using New Vertex button.

If you are using an existing template from the 'Template' drop down list then parameter values for master path and all subparts will be filled in automatically. You can make any changes of predefined parameters or create a new Pwire from scratch, without using a template.

have to select the subpart type: Enclosure Path, Offset Path, or Subrectangle. Then set up for a specific selected subpart type parameter and click Add button. A new subpart will be added to the current Pwire and the subpart list will be updated. Click Ok button to apply changes to the subparts, and Cancel button to discard any changes.

You can create the following types of Pwire subparts.

Enclosure path is a wire with vertices coincident with vertices of the master wire and wire width calculated by formula

    Width = MasterWireWidth – (2 * Enclosure),

Where the Enclosure is defined in 'Enclosure' field from 'Subparts' dialog panel.

You have to choose the layer (drop down list 'Layer') for the Enclosure path. In addition you can specify how

Separation	Center	Left	Right
0	Vertices coincide	Left edge master wire coincide with right edge of offset wire	Right edge master wire coincide with left edge of offset wire
>0	Shift left (separation value is distance between centerlines))	Shift left (separation value is distance between left edge of master wire and right edge of offset wire)	Shift right (separation value is distance between right edge of master wire and left edge of offset wire)
<0	Shift right (separation value is distance between centerlines)	Shift right (separation value is distance between left edge of master wire and right edge of offset wire)	Shift left (separation value is distance between right edge of master wire and left edge of offset wire)

Table 1. Offset path separation

Separation	Center	Left	Right
0	Box center on master wire centerline	Left edge master wire coincide with right edge of box	Right edge master wire coincide with left edge of box
>0	Shift left (separation value is distance between box center and master wire centerline)	Shift left (separation value is distance between left edge of master wire and right edge box)	Shift right (separation value is distance between right edge of master wire and left edge of box)
<0	Shift right (separation value is distance between box center and master wire centerline)	Shift right (separation value is distance between left edge of master wire and right edge of box)	Shift left (separation value is distance between right edge of master wire and left edge of box)

Table 2. Subrectangle separation

the starting and ending points of the Enclosure path are located with respect to the starting and ending points of the master wire. Use 'Begin Offset' and 'End Offset' fields from 'Subparts' dialog to change beginning and ending points of the enclosure path. Starting and ending edges of the enclosure path are extended by specified values if the offset values are positive, and truncated if negative.

Offset path is a wire with edges parallel to the master wire edges and a shift defined by 'Separation' and 'Justification' parameters (see Table 1).

The width of the offset wire is defined by 'Width' parameter. Use drop down list 'Layer' from 'Subparts' dialog to set the layer for the offset wire. In addition you can specify 'Begin Offset' and 'End Offset' parameters to extend or truncate the beginning and ending wire segments correspondingly (see description above).

Subrectangle is the set of boxes lying on the specified layer. The size of the box is defined by 'Width' and 'Length' parameters, where box width is measured across the master wire and the box length along the master wire centerline. The distance between boxes are defined by 'Space' and 'Gap' parameters. Space defines minimal distance between boxes. Gap can be one of two types: Minimum or Distribute. Minimum means that distance between boxes is fixed and equal to space parameter value. The remaining space after the last placed box if there is one is left empty. If you chose Distribute Gap then unused space is evenly distributed between placed boxes, and actual space between boxes is in the range:

```
space value < actual space
< space value + box length.
```

Parameters 'Separation' and 'Justification' is used to place boxes with offset from master wire (see Table 2).

Xi command PWIRE

You can use Xi command PWIRE to create complex object Pwire. Xi command PWIRE has the following format

```
PWIRE {vertices} [/name = pwirename]
[/template = pwiretemplatename]
[/layer = layrname] [/width = value]
[/extend | /flushend | /roundend] [/
extjoint | /filljoint | /roundjoint |
/miterjoint]
[/centeroffset | /leftoffset | /
rightoffset]
[/inwires = {inwires}] [/offwires =
{offwires}] [/boxes = {boxes}]
```

Here {vertices}, /layer, /width, and wire end and joint style arguments are similar to arguments for Xi command 'WIRE' and specify vertices, layer and other wire attributes for master wire of Pwire object. Additional arguments are to define the specific pwire parameters.

Named argument /name specifies a unique name for the pwire object.

Named argument /template allows you to use pwire templates stored in the technology.

Named arguments /inwires, /offwires, /boxes describe pwire subparts (see Xi command PWIRE_TEMPLATE for detailed description of arguments).

Suppose we have Pwire template "GuardRing_2" stored in the technology. Than the following Xi command will create a new Pwire object in the current cell.

```
pwire {-10,-10, -10,20, 20,20, 20,-10, -8,-
10} /template = "GuardRing_2";
```

Storing Pwire templates in the *Expert* Technology

A set of predefined Pwire parameters can be stored in the technology with a unique name and used for quick creation of Pwire objects. Use Setup>>Technology>>Pwire Templates command to add, remove, or modify Pwire templates. Xi command PWIRE_TEMPLATE is also for adding pwire templates to your technology. The command has format

```
PWIRE_TEMPLATE pwire_template_name
[/layer = layername] [/width = value]
[/extend | /flushend | /roundend] [/
extjoint | /filljoint | /roundjoint |
/miterjoint]
[/centeroffset | /leftoffset | /
rightoffset]
[/inwires = {inwires}] [/offwires =
{offwires}] [/boxes = {boxes}],
```

where named arguments /inwires, /offwires, and /boxes describe enclosure paths, offset paths, and subrectange subpart of pwire object. There are set of predefined xi attributes used for definition of pwire subparts, such as:

PWA_LAYER	defines subpart layer;
PWA_WIDTH_DELTA	defines Enclosure path width;
PWA_WIDTH	defines Offset path or Subrectangle witdh;
PWA_LENGTH	defines Subrectangle length;
PWA_OFFSET	defines offset path separation from master wire;
PWA_OFFSET_TYPE	defines offset separation type;
PWA_MIN_GAP	defines space between boxes for subrectangle subpart;
PWA_BOX_DISTR	defines box distribution for subrectangle subpart;

Each subpart is described as a sequence of attributes with values of subparts parameters. For example, the following Xi command adds Pwire template "GuardRing_2" with two enclosure paths and two subrectangles.

```
wire_template "GuardRing_2" /layer =
"METAL1" /width = 2 /offset = 1
/inwires = ({{attr_create(PWA_LAYER,"METAL2"),
      attr_create(PWA_WIDTH_DELTA, 0.2)},
      {attr_create(PWA_LAYER, "NWELL"),
      attr_create(PWA_WIDTH_DELTA, 0.5)}})
/boxes = ({{attr_create(PWA_LAYER, "CNTG"),
      attr_create(PWA_WIDTH, 1),
      attr_create(PWA_LENGTH, 1),
      attr_create(PWA_OFFSET, 0),
      attr_create(PWA_OFFSET_TYPE, POT_CENTER),
      attr_create(PWA_MIN_GAP, 1),
      attr_create(PWA_BOX_DISTR, BDT_EVEN)},
      {attr_create(PWA_LAYER, "CONT"),
          attr_create(PWA_WIDTH, 0.5),
      attr_create(PWA_LENGTH, 0.5),
      attr_create(PWA_OFFSET, 0),
      attr_create(PWA_OFFSET_TYPE, CENTER),
      attr_create(PWA_MIN_GAP, 2),
      attr_create(PWA_BOX_DISTR, BDT_EVEN)}});
```

Pwire template can be stored in and loaded from the *Expert* external ASCII *.tcn technology file (see example below).

```
PwireTemplate
  {
  PwireTemplateName = "GuardRing_1"
  MasterWireLayer = "METAL1"
  MasterWireWidth = 2
  MasterWireJoint = EXTEND
  MasterWireEnd = EXTEND
  MasterWireOffset = 0
  MasterWireOffsetType = CENTER
  PwInWire
    {
    PwLayer = "METAL2"
    PwBeginOffset = 0
    PwEndOffset = 0
    PwWidthDelta = 0.2
    }
  PwInWire
    {
    PwLayer = "NWELL"
```
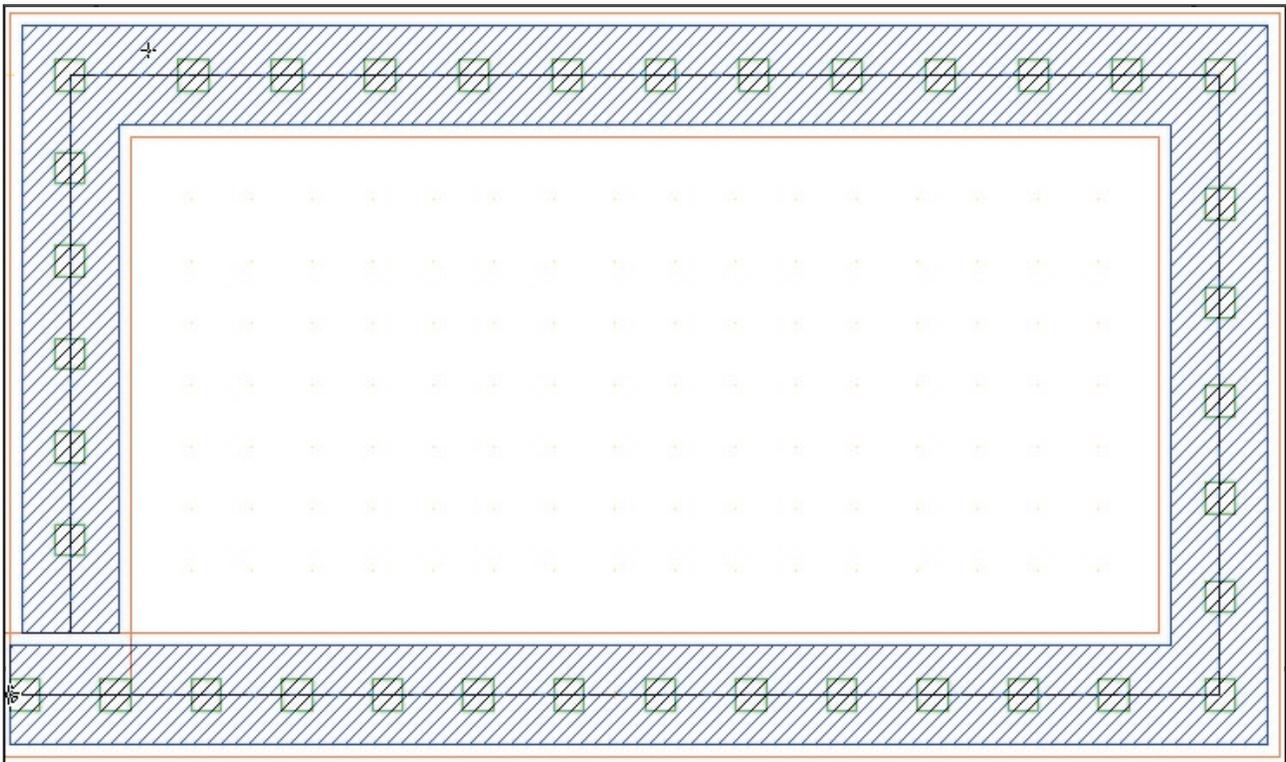
Figure 3. Example of Pwire.

```
    PwBeginOffset = 0

    PwEndOffset = 0

    PwWidthDelta = 0.5

    }

  PwBox

    {

    PwLayer = "CNTG"

    PwBeginOffset = 0

    PwEndOffset = 0

    PwWidth = 1

    PwLength = 1

    PwOffsetType = CENTER

    PwDistrType = EVEN

    PwOffset = 0

    PwMinSpace = 1

    }

  PwBox

    {

    PwLayer = "CONT"

    PwBeginOffset = 0
```

```
    PwEndOffset = 0

    PwWidth = 0.5

    PwLength = 0.5

    PwOffsetType = CENTER

    PwDistrType = EVEN

    PwOffset = 0

    PwMinSpace = 2

    }

}
```

Example of Creating a Pwire

The Pwire with parameters shown on Figure 1 and Figure 2 looks like this in Figure 3.

Conclusion

Expert Pwire engine is a convenient and powerful feature of the layout editor. It speeds up and improves the quality of IC layout designs.