8

Power Supply

8.1 Introduction

During the specification of a circuit the designer does not enter the power supply nets. These can be generated automatically by an appropriate tool ([Sch92]) which has to perform the following two tasks:

- calculation of the topology and
- sizing of the wiring segments

The calculation of the topology is only possible, if all basic cells in the design have pins for the connection of the power supply nets. In our designs from the editor session, where we used the CADIC basic cell library, these pins are denoted by small blue squares at the cell borders, they are labelled with the names `VDD` and `VSS`.

In today's fabrication process both supply nets will be layed out in the same metal layer. This implies that the nets must be free of any crossings. The algorithm in CADIC achieves such a layout by a tree like structure for both nets which can be derived directly from the topology of the graphical specification.

Beside the calculation of the topology of the supply nets, the wiring segments of these nets have to be sized according to power consumption aspects. The sizing has to guarantee that the power drop from the pads at the border of the whole chip to the inner pins at the basic cells is smaller that a

given bound. At the same time the wire segments should not be oversized because of the limited chip area. Above this there is a minimum wire width which has to be respected. This means, that we have to regard different optimization goals which are influencing each other. If we choose a certain layout for the wires this implies their width and the power consumption. A certain wire width has a backdraw to the wire layout because of geometrical rules.

8.2 The Algorithm integrated in CADIC

The tool which is integrated in CADIC , generates the power supply nets in two independent steps. First it calculates the topology of the nets according to the graphical specification. In a second step the sizing for the nets is calculated such, that the power consumption and the chip area for the wires are balanced within the given topology.

8.2.1 Calculation of the Topology

As almost all integrated algorithms in CADIC the calculation of the topology for the power supply nets is done in a hierarchical manner. The algorithm constructs the nets in a bottom–up process following the DAG structure of the given circuit. This means, that for each treenode the power supply nets are created once. The location of the pads for these nets on the corresponding schematic borders are used on the next higher hierarchy level as the new starting points. For the macro cells on each hierarchy level the algorithm creates new pins for the connection of the power supply nets.

The underlying algorithm ([Kol86]) is based on the representation of each treenode by a bicategorial expression as we have introduced it in section 4.2. Before the topology is created, for each treenode a data structure for the bicategorial expression is generated by a slicing algorithm. This algorithm uses the graphical specification to split it according to the operations $\ominus$ and $\oplus$. This algorithm is stored in an appropriate function library, which can be used for other tools using the bicategorial representation. Another

example for such a tool is the place&route algorithm which will be explained in chapter 9.

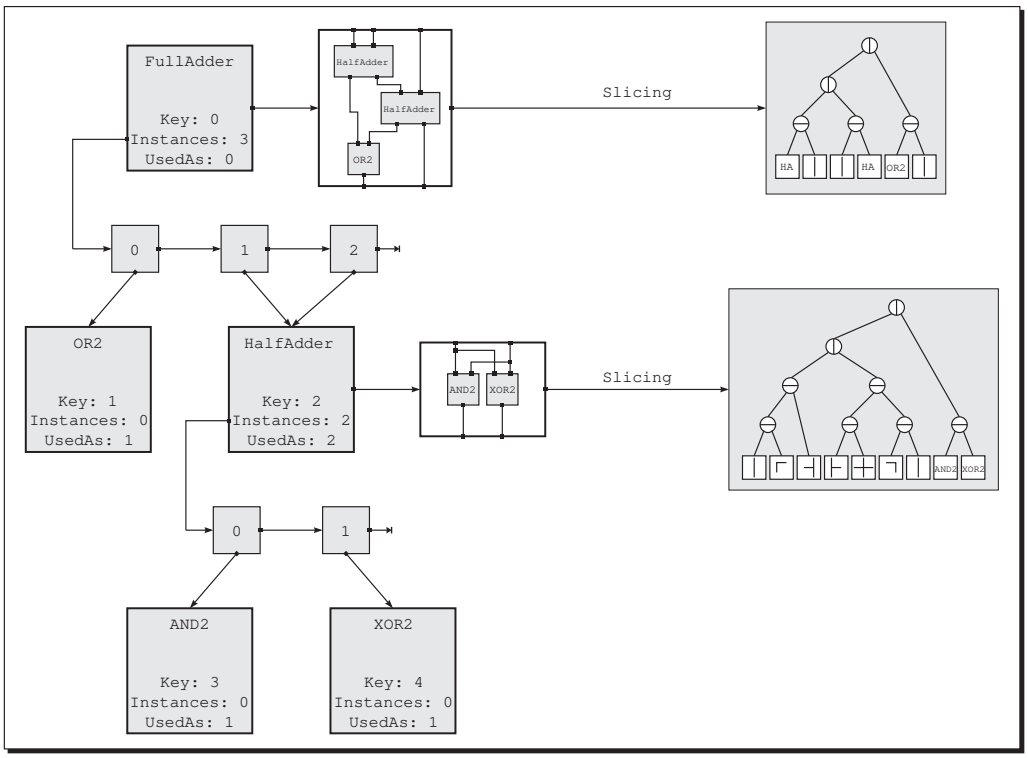


Figure 8.1: Generation of a slicing tree for each treenode in a hierarchical description

In figure 8.1 it is shown, how the representation in form of a bicategorial expression is created for each treenode in the hierarchy. By applying the slicing algorithm for each treenode a pointer to the data structure for the bicategorial expression is set. Each expression is represented by a tree like structure, where the leaves of the tree are built of nodes, edges and instances in the corresponding Cgraph. The inner nodes of the data structure, which is called the *syntax tree*, are operational nodes representing the concatenations of the corresponding subtrees.

On each hierarchy level the integrated algorithm works bottom–up in the syntax tree, where the operations at the inner nodes imply the local topology of the supply nets ([Kol86]). To create an easy to handle result, we assume that each treenode and each instance has exactly one `VDD` and one `VSS`

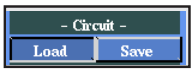
pad. The VDD (VSS) pins and pads are always located at the northern or the eastern (the southern and the western) border. This implies a simple method to create the two power nets without crossings.

8.2.2 Sizing of the Power Supply Nets

After the topology of the power supply nets is generated each wire segment of these nets is sized according to the power consumption of the connected elements. The sizing algorithm is also working hierarchically on the underlying DAG structure. It is based on heuristics which are explained in detail in [Kol86]. The calculated sizing results are stored as attributes to the created wiring edges in the corresponding Cgraphs, such that they can easily be displayed in the graphical specification.

8.3 Power Nets for the Fulladder

Load Circuit



Before we can construct the power supply nets for a circuit, you must load this circuit first. After pressing the push button `Load` in the submenu `-Circuit-`, a list window pops up onto the work area which contains the list of all existing circuits. Move the cursor near to the entry `FullAdder` and select by pressing the left mouse button.

Create the Topology of the Power Nets

After you have loaded the hierarchy for `FullAdder` you can create the topology for the power supply nets. You activate the algorithm by selecting the entry `Create` from the submenu `-Power-Tree-`. Then you can read the following informations in the message window:

```
Generate the Power-Tree for HalfAdder
Generating/Reading Slicing(HalfAdder)
Routing power_wire(HalfAdder)
Building Power-treenode(HalfAdder)
```

```
Generate the Power-Tree for FullAdder
Generating/Reading Slicing(FullAdder)
Routing power_wire(FullAdder)
Building Power-treenode(FullAdder)
```

These messages illustrate the bottom–up working method of the algorithm. First the power supply nets for the subcircuit `HalfAdder` are generated. For this purpose the algorithm creates or simply reads the slicing of `HalfAdder` as the second message line implies. If the graphical representation of `HalfAdder` has not changed since the last call of the slicing procedure, it can read the old information instead of calculating it from the scratch.

With the help of the slicing information the power supply nets are routed through the Cgraph of the corresponding hierarchy level (illustrated by the message `Routing power_wire(HalfAdder)`). In the final step for this hierarchy level the power nets are connected to the border of the schematic and the pads for the external connections to these nets are created.

Now that the power supply nets for `HalfAdder` exist, the nets for the next higher level `FullAdder` can be generated. Again the slicing for this level is created or read form the directory given by the environment variable `SLICEDIR` and then the power nets are routed through the Cgraph of `FullAdder`. The resulting topology of the supply nets can be seen in figure 8.2.

Note that the instances for `HalfAdder` have two new pins which are labelled `VDD` and `VSS`. These pins correspond to the pads created at the lower hierarchy level, where the supply nets are connected to the border of the schematic. As mentioned above the bottom–up working method of the algorithm implies that for all subcircuits at the current hierarchy level, the power supply nets must have been created, such that the power pins at all instances exist.

For the basic cells at the lowest hierarchy level the supply pins are given by the information from the cell library. An important condition is that the pins for `VDD` are always at the northern or the eastern border and the

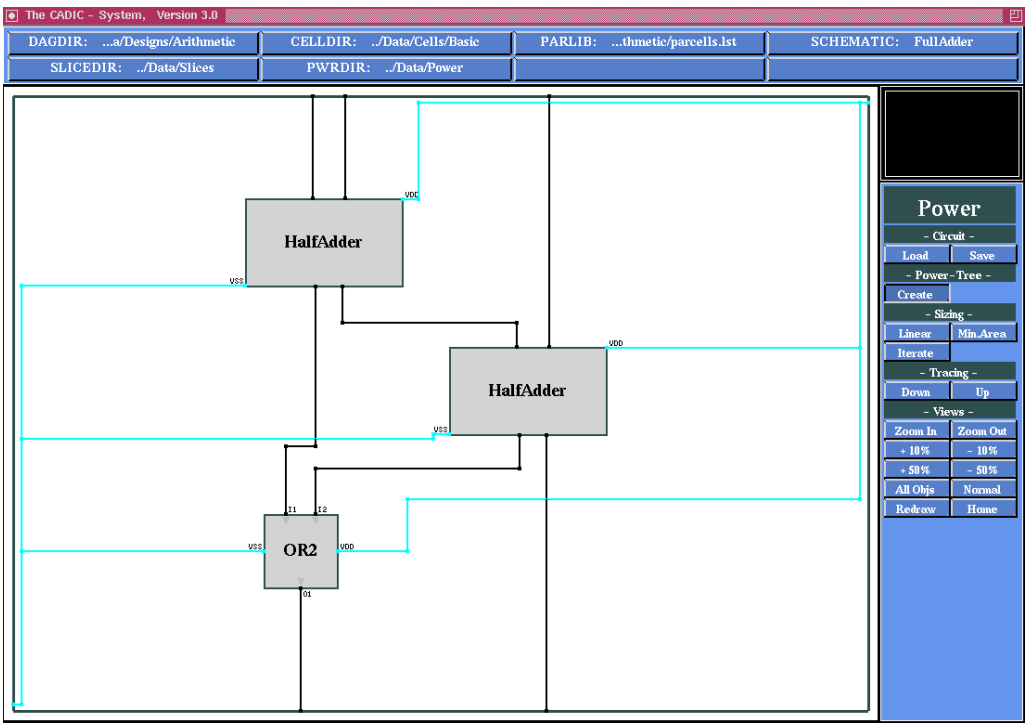
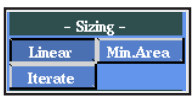


Figure 8.2: Topology of the power supply nets for the `FullAdder` level

pins for `VSS` are at the southern or the western border. For the macro cells this is automatically performed by the power supply algorithm, but for the basic cells this must be regarded during the design of a new cell library.

Sizing of the Supply Nets



After the topology for the supply nets has been created, the wire segments of these nets can be sized according to the power consumption rules. For this purpose CADIC offers three different heuristics which work on the hierarchical representation of the circuit. In our session we will use the linear sizing heuristic which can be activated by selecting the entry `Linear` from the submenu `-Sizing-`.

The results of the sizing heuristic are displayed by small labels at each wire segment of the power supply nets (c.f. figure 8.3). The width of each segment is given in μm, where the minimum width of a wire in CADIC 's layout system is 2μm.

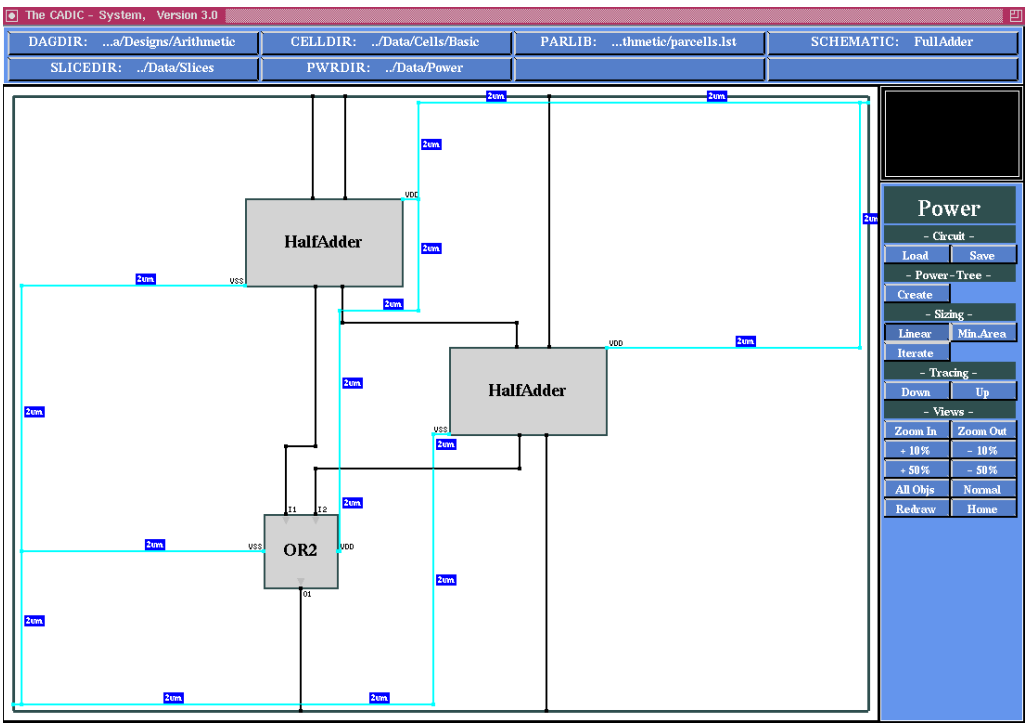
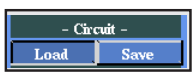


Figure 8.3: Sizing of the wire segments of the power supply nets for the `FullAdder`

In our small example of the fulladder, the sizing heuristic sets all wire widths to the same value of 2μm. Again you can inspect the results of the algorithm at the lower hierarchy level with the help of the tracing functions.

8.4 Power Supply for the Conditional Sum Adder

Load Circuit



Before we can create the power supply nets for the conditional sum adder, we have to load an element of the parameterized description. Select the entry `Load` from the submenu `-Circuit-` and choose `CSA[n]` from the schematic list window. For the parameter value n type in `16` at the corresponding input window.

Create the Topology of the Supply Nets

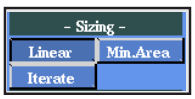
If you activate the entry `Create` in the `-Power-Tree-` submenu the topology for the supply nets on each hierarchy level of `CSA[16]` is created. In the message window the algorithm displays the informations for each treenode:

```
Generate the Power-Tree for CSA[1]
Generating/Reading Slicing(CSA[1])
Routing power_wire(CSA[1])
Building Power-treenode(CSA[1])

Generate the Power-Tree for SEL[1]
Generating/Reading Slicing(SEL[1])
Routing power_wire(SEL[1])
Building Power-treenode(SEL[1])
...
Generate the Power-Tree for SEL[9]
Generating/Reading Slicing(SEL[9])
Routing power_wire(SEL[9])
Building Power-treenode(SEL[9])

Generate the Power-Tree for CSA[16]
Generating/Reading Slicing(CSA[16])
Routing power_wire(CSA[16])
Building Power-treenode(CSA[16])
```

Sizing for the Power Nets



Now we call the sizing heuristic for the power supply nets of the 16 bit conditional sum adder. Again we will use the linear sizing algorithm which can be activated by the entry `Linear` form the submenu `-Sizing-`. After the algorithm has finished you can see the results within the small labels at each segment of the supply nets as it is illustrated in figure 8.4.

In this example the sizing algorithm has to enlarge the wire segments near to the border of the whole circuit. For example the VDD net has the width 22μm at the eastern border of the schematic. The label at the corresponding

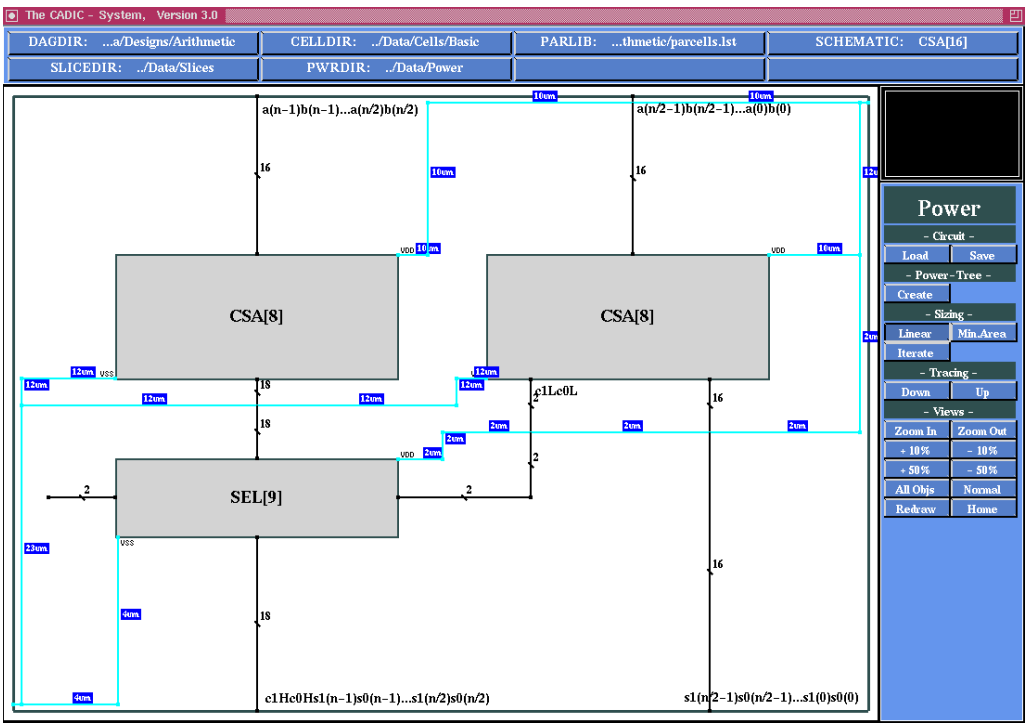


Figure 8.4: Power supply nets for `CSA[16]` with linear sizing of the wire segments

wire segment is not shown, because the length of the segment is too small. Nevertheless you can look at it with the help of the zooming function `Zoom In` from the submenu `-Views-`. After the branching node this wire segment is split into two segments of the width 12μm and 10μm, respectively.

The `VSS` net which starts at the western border of the schematic has the size 27μm. Again you must zoom in to see the label at this short segment. After the branch, it is divided into two segments of width 23μm and 4μm, the first of which is split again into two segments of width 12μm and then connected to the two instances `CSA[8]`.

With the help of the tracing function `Down` from the submenu `-Tracing-` you can follow this wire into the instances `CSA[8]`. Select the left 8 bit conditional sum adder and trace down into it. At this level you can control the connection of the power supply nets to the next subcircuits in the hierarchical description.

Now trace down until you reach the lowest level. For this purpose we select the left instance, the conditional sum adder for the upper half of the operands, at each hierarchy level until we reach the level `CSA[1]` as it is shown in figure 8.5.

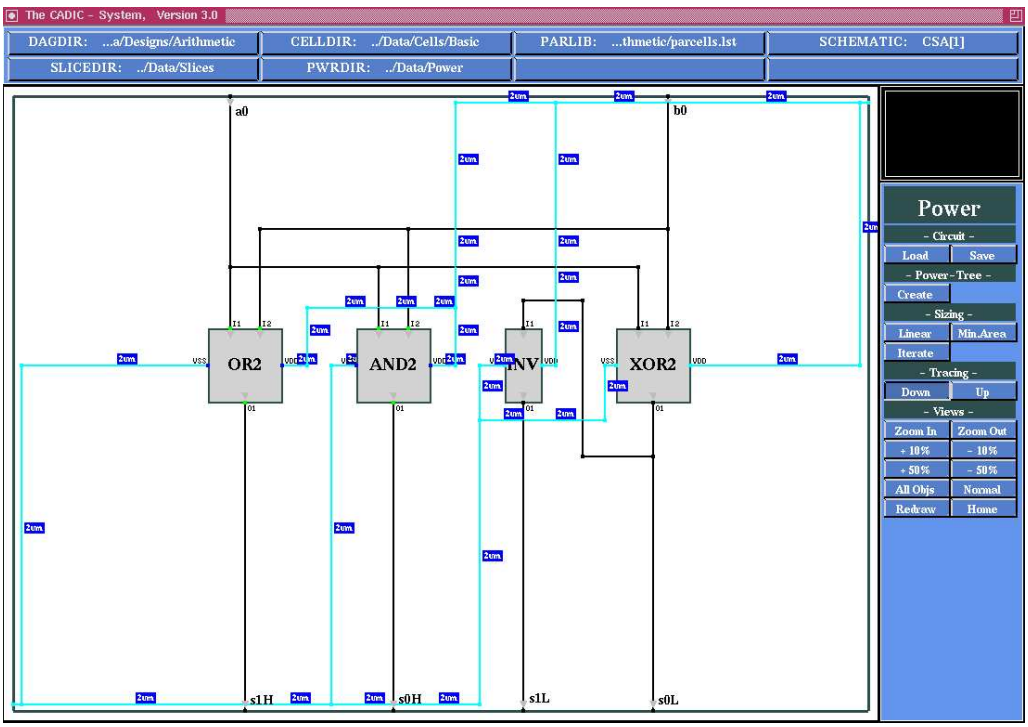


Figure 8.5: Power supply nets for the 16 bit conditional sum adder at the lowest hierarchy level `CSA[1]`

At this level which only contains basic cells, all wire segments are sized to the minimum wire width of 2μm. The width of the wires at higher levels are implied by these basic circuits, because of the bottom–up working method of the sizing algorithm.

Changing the View

With the help of the functions from the submenu `-Views-` you can change the graphical representation of the schematic. The functions in the menu of the power supply tool are identically to those in the hierarchy menu. For an explanation you can look at section 5.9.

To terminate the power supply menu you have to terminate any currently active function first. In most cases this can be done by pressing the right mouse button at most twice within the workarea (see also the description of the appropriate function). After that you are in menu selection mode and you can close the power supply menu by pressing the right mouse button within the menuline. Now you return to the main menu of the CADIC system. From there you can select another tool or you can terminate the whole system call by selecting the entry `Exit` in the main menu.

Normally the layer assignment and the power supply tools are called before you create the final layout of the circuit. When you have created the power supply nets for a circuit you can change to the menu for generating its layout by going to the main menu of CADIC and then to the layout submenu (c.f. chapter 9).