4 Placing and Routing Block Designs

Introduction

L-Edit Block Placement and Routing (BPR) automatically places and routes a design using a TPR or EDIF format netlist and layout blocks as input.

BPR initializes a design by reading the netlist for the blocks to use and for their connectivity. It compares the netlist with the blocks in your layout and places referenced blocks in your design within a top-level cell.

Connectivity is displayed as a network of routing guides for the pin-to-pin connections. Throughout the BPR process, the connectivity display is interactively updated as you move objects and add routing.

Once your design is initialized, you can use BPR to automatically or manually place blocks according to the relative importance of minimizing the total routing length or the total area covered by blocks. BPR can perform incremental placement relative to the netlist, where specific sections of the design can be replaced or rerouted while any unaffected placement or routing is preserved.

BPR supports fully automatic routing or assisted manual routing. During assisted manual routing, BPR guides your cursor from pin to pin as you route your nets. Keyboard shortcuts allow you to quickly select from the defined routing layers, and BPR automatically adds the appropriate via when you change from one routing layer to another.

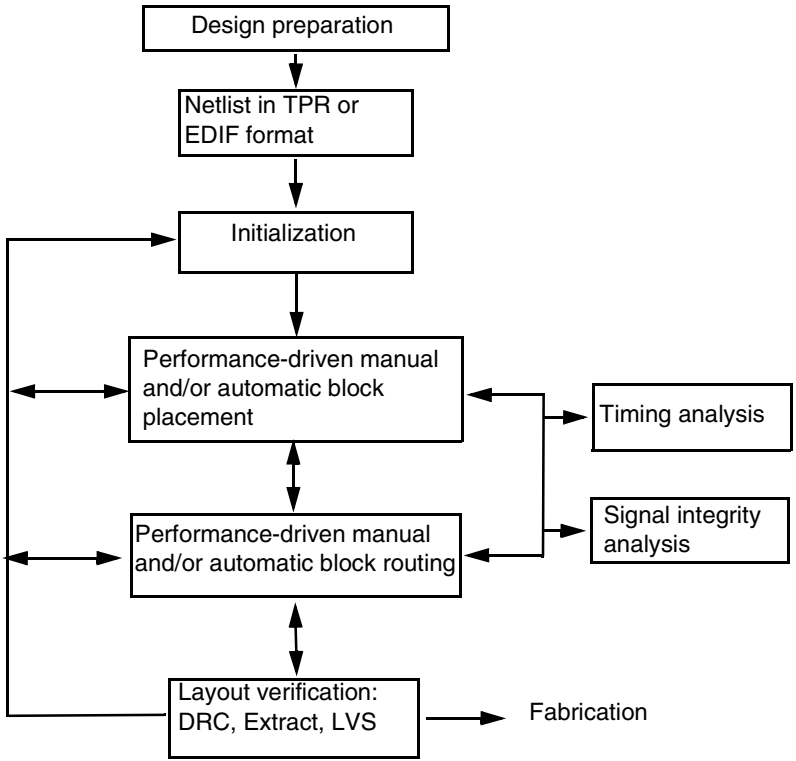
Any BPR design must have one cell defined as the *top-level BPR cell.* This is the cell at the very top of the cell hierarchy for the file, which contains all blocks. There can only be one top-level BPR cell per file, and it is the cell that is active when you initialize your design. However, the top-level BPR cell in a file can cross-reference the top-level BPR cell from another file.

BPR also allows you to specify a *top-level I/O cell*, which is a cell used to define the location of the external pins of the top-level BPR cell and the maximum size of the top-level BPR cell. When you use a top-level I/O cell, all blocks in a design will be placed inside of the top-level I/O cell during initialization, and connectivity will be shown from the blocks to the I/O cell ports.

At any point in the BPR process you can open the **Netlist Navigator** for complete routing, pin, and signal information for each net.

BPR also includes interactive signal integrity analysis and timing analysis tools that can be used at any time during placement and routing to verify that your performance requirements are met.

BPR Design Process



The L-Edit / BPR process consists of five stages, plus optional timing analysis, signal integrity analysis, and layout verification steps.

☑ **Design Preparation**. This stage must be performed before BPR is initialized. During design preparation, you create and/or copy all blocks to be placed and routed during BPR into the current design file.

☑ **Initialization**. During this stage you specify the netlist file L-Edit will read, which cell will be used as the top-level BPR cell, the net names associated with any special signal types, and whether a top-level I/O cell will be used.

At the end of initialization, instances of all cells used in the netlist are placed in the layout and the netlist connectivity is shown as routing guides on the routing guide layer. Use **Tools > BPR > Initialization** (page 2-174) to specify initialization parameters.

☑ **Placement**. During placement, blocks are manually and/or automatically positioned in the layout. You can reposition cells in conjunction with analysis of the electrical and physical properties of the design. Connectivity display is constantly updated. Use **Tools > BPR > Automatic Placement** (page 2-191) to specify automatic placement parameters.

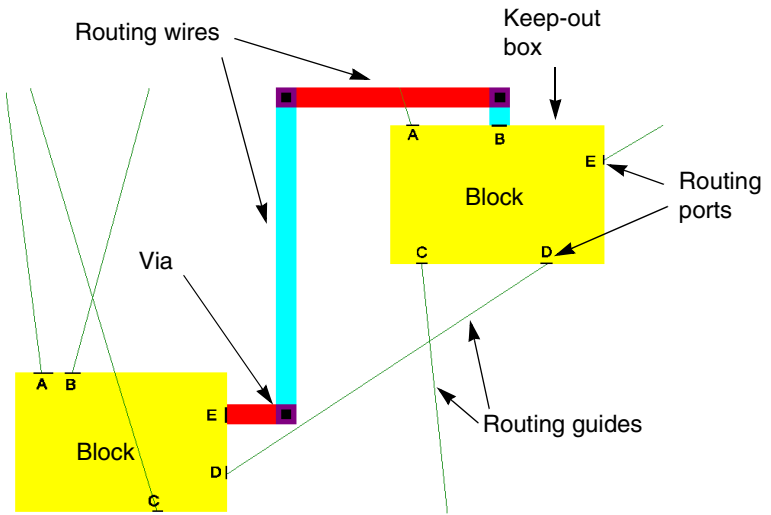
☑ **Routing**. Routing is the creation and spatial arrangement of interconnections (wires and vias) between blocks and, optionally, I/O pads. Routing can be performed either automatically or using the connectivity guided assisted manual router. BPR will preserve unaffected routing when the placement of selected blocks changes. Parameters for routing are specified using **Tools > BPR >**

Setup—Manual Router (page 2-217) and **Tools > BPR > Setup—Autorouter** (page 2-206).

☑ **Timing Analysis and Signal Integrity Analysis**. L-Edit provides electrical analysis tools for any stage of the BPR process that simulates the signal performance of single or multiple nets. Performing these analyses interactively with placement and routing allows you to evaluate the impact of your chosen placement, interconnect topology, and driver and receiver parameters on signal integrity and timing parameters of the design. (See Running Timing Analysis on page 2-243 and Running Signal Integrity on page 2-250.)

☑ **Layout Verification**. Layout verification is accomplished by invoking the Design Rule Checker (DRC) (see Checking Design Rules on page 3-10), the general device extractor (Extract) (see Extracting Layout on page 3-84), and LVS, the Tanner EDA netlist comparison tool (see Getting Started with LVS on page 3-143).

BPR Glossary



Blocks and other electrically connected objects shown in this figure are defined below.

block

A cell that is electrically connected by being referenced in the netlist (see figure above). Following initialization, blocks are shown with routing guides connecting their routing ports.

connection

A net section from one pin to another.

electrically connected object

A block, routing wire, via, or routing guide in the BPR design (see figure above.)

keep-out

A box on the keep-out layer that designates an area where the automatic router will not route. Only a box will be recognized as a keep-out—any non-box on the keep-out layer will be ignored.

MBB

The minimum bounding box, which is the smallest box that contains a set of objects.

non-electrically connected object

Any object in the design—box, polygon, wire, circle, ruler, port, instance, etc.—other than a block, route, via, or routing guide.

route

A combination of routing wires, routing guides, and vias connecting one routing port to another.

routing guide

A non-editable wire on the routing guide layer that indicates the unrouted part of a connection (often referred to as *rat's nest*).

routing port

A port on a cell indicating the location of a pin needed for routing. This port designates where a route will start or end.

routing wire

A wire with associated connectivity information. A routing wire can comprise any routing section from via-to-via to an entire net.

subcircuit recognition polygon (SRP)

A box or 90° polygon on the layer used for extraction, which marks a cell as a subcircuit cell and delimits the area of the cell. The subcircuit

recognition polygon also indicates the perimeter on which routing ports must be placed.

top-level BPR cell

The cell at the highest level of the current design hierarchy. There can only be one top-level BPR cell per file.

top-level I/O cell

A cell containing ports that will be defined as the external pins of the top-level BPR cell. When this cell is used in a design, all blocks will be placed inside of and connected to this cell during initialization.

via

A cell used to indicate a connection from one routing layer to another.

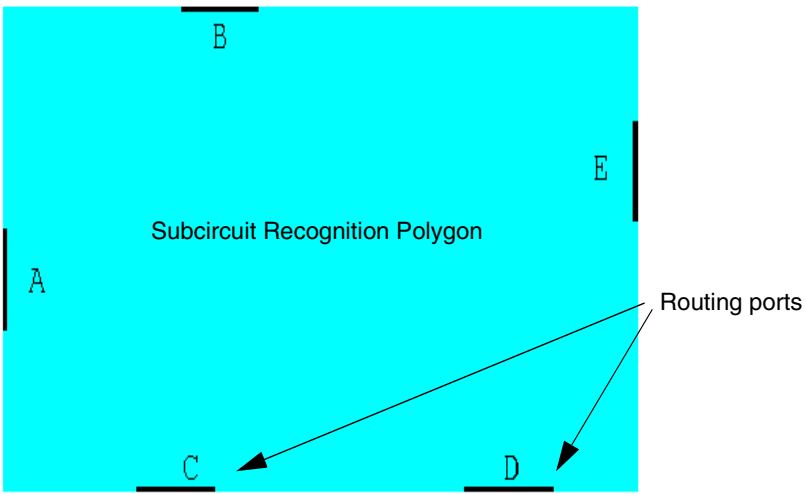
Design Preparation

In order to initialize a design for BPR, you must first:

- Define a technology setup appropriate to your design. If you start L-Edit with an empty design file, use **File > New** to copy the TDB setup from a design or technology file that already contains the correct setup.
- Create or copy into your design file (**.tdb**) each of the blocks and cells referenced in the netlist used to initialize your design.
- Create a netlist file that will be used during initialization. This netlist contains a description of your schematic and identifies the blocks and other cells required for placement and routing.

Defining Blocks

A *block* is an instance of a cell that is electrically connected by being referenced in the netlist. Any L-Edit cell that can be referenced in other cells as an instance and has ports can be used as a block in BPR.



Typical BPR block with routing ports and a subcircuit recognition polygon

At initialization, all instances of the same cell must have a unique instance name both in the netlist and the layout. Instance names can be changed in the layout, but BPR cannot initialize a design with non-unique instances.

Keep-outs are not required, but may be added to define areas where automatic routing will not be allowed. (See Routing Keep-Outs, below.)

Following initialization, blocks are displayed with routing guides connecting their routing ports. (See Routing Guides on page 2-188.)

Routing Keep-Outs

Keep-outs are used to define areas where the BPR automatic router cannot route. Keep-out areas for a block may be designated either explicitly on the keep-out layer or implicitly using a subcircuit recognition polygon. A keep-out will exclude routing on both automatic routing layers.

All boxes on the keep-out layer will be interpreted as keep-outs. Any non-boxes on the keep-out layer will be ignored.

The automatic router is a gridded router that uses the center of a routing wire for positioning, so routing wires can overlap a keep-out boundary. If you use the extend wire end style, a portion of the routing wire can extend into a keep-out region at the port.

Note: Keep-out boundaries should be drawn on the routing grid. Otherwise, it is possible for the router to extend into the keep-out area by one routing grid dimension.

Boxes on the subcircuit recognition layer are implicit keep-outs. Non-boxes on the subcircuit recognition layer do not function as keep-outs. A subcircuit recognition polygon must be associated with a block to be treated as a keepout.

Routing Ports

Routing ports for a block are defined by being referenced in the netlist. You create routing ports to indicate pin locations to the automatic router and the assisted manual router. A routing port for BPR should be a point, but may be a line.

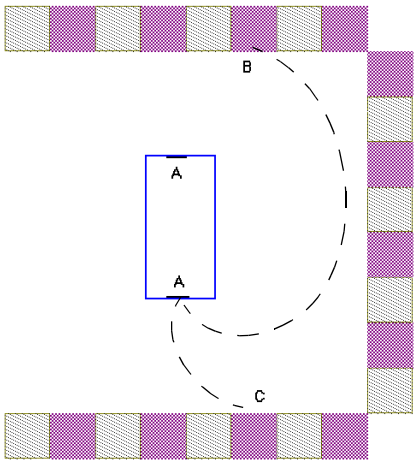
BPR will only route to a port on a defined routing layer. To be able to route to a pin using any of the defined routing layers, you must create a port on *each* routing layer. The multiple ports that define the different routing layers of a routing pin must have the same size and shape, and they must be in the same location.

Note that a routing port in a block can be moved or deleted. However, L-Edit will provide a warning if you try either operation.

The BPR autorouter will route to any referenced port inside a block that is on the boundary of, or entirely outside of, a keep-out box or SRP. Ports entirely inside of the keep-out or SRP may be routed to if they are less than one routing grid dimension from the boundary.

It is possible to have routing ports inside a block, but to avoid design rule violations you must add the appropriate keep-outs to allow the automatic router to route to these routing ports.

If the same layer contains more than one port with the same name in different locations, during initialization BPR will simply create a routing guide to the first port it finds and ignore all others. For this reason, you should avoid cells with different possible routing entry locations for pins with the same name.



BPR initialization will create a routing guide to the first pin labeled **A** that it finds in the net connecting pins **A**, **B**, and **C**.

Top-Level BPR Cell

The top-level BPR cell is the cell at the highest level of the cell hierarchy for a design. It contains all blocks in the design, the routing connectivity, and the top-level I/O cell, if one is used. There can only be one top-level BPR cell per file. A top-level BPR cell may be deleted

L-Edit makes the cell that is active when you invoke BPR initialization the top-level BPR cell for a design. However, you cannot use your top-level I/O cell as a top-level BPR cell.

Top-level I/O Cell

The *top-level I/O cell* is a cell used to define the external routing ports for a design. Port names in the top-level I/O cell must be unique unless they are to be used for multiple-layer routing. Ports for multiple-layer routing must have the same location, size, and shape on each routing layer.

During initialization, all blocks in a design will be placed inside of and connected to the top-level I/O cell when the **Top-level I/O cell** option is checked in the **BPR Initialization** dialog.

This cell can be copied, but it cannot be moved or deleted unless the top-level BPR cell is deleted. Individual ports may be repositioned within the cell. Note that L-Edit will not provide a warning when a routing port is moved or deleted.

Because BPR routing can only take place in L-Edit's positive coordinate space, initialization will position the top-level I/O cell in the upper-right quadrant of your design.

Specifying External Routing Ports

BPR provides several methods for specifying the name and position of external routing ports.

Defining Ports Using Pad Definitions in the Netlist

If pads are defined (for example, when you are using a padframe), you can use a **CP** statement in the TPR netlist or **PAD** properties in the EDIF netlist to reference the location and pad name in the layout (see TPR Files on page 2-337 and EDIF Files on page 2-326).

Defining Ports for a TPR Netlist

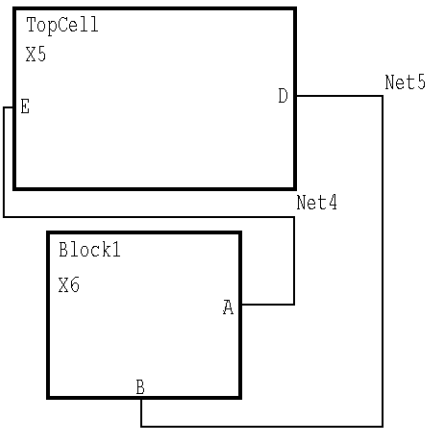
When BPR is initialized with a TPR netlist, you need to create an I/O cell in your schematic and explicitly include it in the netlist. Or, if an I/O cell is not defined in your schematic, you can define port names with the same names as the net names defined for the I/O cell in your design. BPR then will automatically add connections from the I/O cell ports that match net names to the nets.

Defining Ports for an EDIF Netlist

When BPR is initialized with an EDIF netlist, you do not need to define an I/O cell in your schematic as long as external ports for the top-level BPR cell are

defined in your netlist. Or, as with TPR format, you can create an I/O cell in the schematic and include it explicitly in the netlist.

The figure below shows the schematic of a 1-bit adder block to be created using BPR.



The EDIF (**.edn**) netlist for this schematic, shown below, indicates that five blocks are to be placed and routed together. Five pins in the netlist are designated as external pins of the block to be created.

```
(edif
          Adder1Bit
          (edifVersion 2 0 0)
          (edifLevel 0)
          (keywordMap (keywordLevel 0))
          (status
            (written
                  (timestamp 1999 2 11 20 55 6)
                  (program "S-Edit" (version "Version 6.00 Beta 6"))))
          (library Adder1Bit
            (edifLevel 0)
            (technology (numberDefinition (scale 1 (E 1 -12) (unit CAPACITANCE))))

            (cell XOR2
                  (cellType GENERIC)
                  (status
                        (written
                              (timestamp 1999 2 11 20 55 6)
                              (program "S-Edit" (version "Version 6.00 Beta 6"))))
                  (view view_1
                        (viewType NETLIST)
                        (interface
                              (port A
                                  (property (rename &_23 "#") (string "1"))
                                  (direction INPUT))
                              (port B
                                  (property (rename &_23 "#") (string "2"))
```

```
                        (direction INPUT))
                    (port Out
                        (property (rename &_23 "#") (string "3"))
                        (direction OUTPUT))
                    (property DEVICE (string "XOR2"))
                    (property LEVEL (string "STD"))
                    (property PARTS (string "1"))
                    (port Gnd
                        (property (rename &_23 "#") (string "4"))
                        (property implicitPortClass (string "Gnd"))
                        (property portType (string "supply")))
                    (port Vdd
                        (property (rename &_23 "#") (string "5"))
                        (property implicitPortClass (string "Vdd"))
                        (property portType (string "supply")))
)))

(cell NAND2
      (cellType GENERIC)
      (status
            (written
                    (timestamp 1999 2 11 20 55 6)
                    (program "S-Edit" (version "Version 6.00 Beta 6"))))
      (view view_1
            (viewType NETLIST)
            (interface
                    (port A
                        (property (rename &_23 "#") (string "1"))
                        (direction INPUT))
                    (port B
                        (property (rename &_23 "#") (string "2"))
                        (direction INPUT))
```

```
                        (port Out
                            (property (rename &_23 "#") (string "3"))
                            (direction OUTPUT))
                        (property DEVICE (string "NAND2"))
                        (property LEVEL (string "STD"))
                        (property PARTS (string "1"))
                        (port Gnd
                            (property (rename &_23 "#") (string "4"))
                            (property implicitPortClass (string "Gnd"))
                            (property portType (string "supply")))
                        (port Vdd
                            (property (rename &_23 "#") (string "5"))
                            (property implicitPortClass (string "Vdd"))
                            (property portType (string "supply")))
))

(cell Adder1Bit
      (cellType GENERIC)
      (status
            (written
                    (timestamp 1999 2 11 20 55 6)
                    (program "S-Edit" (version "Version 6.00 Beta 6"))))
      (view view_1
            (viewType NETLIST)
            (interface
                    (port A
                        (property (rename &_23 "#") (string "1"))
                        (direction INPUT))
                    (port B
                        (property (rename &_23 "#") (string "2"))
                        (direction INPUT))
                    (port Cin
```

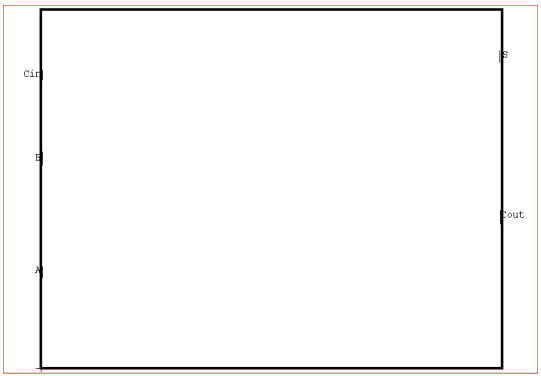
```
            (property (rename &_23 "#") (string "3"))
            (direction INPUT))
        (port Cout
            (property (rename &_23 "#") (string "4"))
            (direction OUTPUT))
        (port S
            (property (rename &_23 "#") (string "5"))
            (direction OUTPUT))
)
(contents
        (instance X1
            (viewRef view_1 (cellRef XOR2))
            (portInstance A)
            (portInstance B)
            (portInstance Out)
            (portInstance Gnd)
            (portInstance Vdd)
            (property EDIF_PRIMITIVE (string "0"))
            (property Instance (string "X1"))
        )
        (instance X3
            (viewRef view_1 (cellRef XOR2))
            (portInstance A)
            (portInstance B)
            (portInstance Out)
            (portInstance Gnd)
            (portInstance Vdd)
            (property EDIF_PRIMITIVE (string "0"))
            (property Instance (string "X3"))
        )
        (instance X2
            (viewRef view_1 (cellRef NAND2))
```

```
        (portInstance A)
        (portInstance B)
        (portInstance Out)
        (portInstance Gnd)
        (portInstance Vdd)
        (property EDIF_PRIMITIVE (string "0"))
        (property Instance (string "X2"))
    )
    (instance X4
        (viewRef view_1 (cellRef NAND2))
        (portInstance A)
        (portInstance B)
        (portInstance Out)
        (portInstance Gnd)
        (portInstance Vdd)
        (property EDIF_PRIMITIVE (string "0"))
        (property Instance (string "X4"))
    )
    (instance X5
        (viewRef view_1 (cellRef NAND2))
        (portInstance A)
        (portInstance B)
        (portInstance Out)
        (portInstance Gnd)
        (portInstance Vdd)
        (property EDIF_PRIMITIVE (string "0"))
        (property Instance (string "X5"))
    )
    (net Cout
        (joined
            (portRef Out (instanceRef X5))
            (portRef Cout)
```

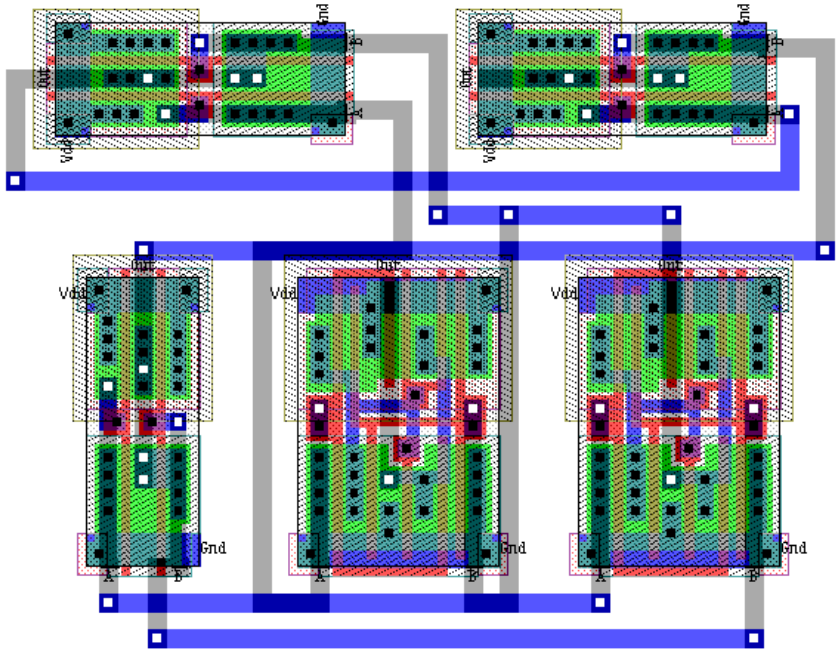
```
))
(net Net1
    (joined
        (portRef B (instanceRef X3))
        (portRef Out (instanceRef X1))
        (portRef B (instanceRef X4))
))
(net Net2
    (joined
        (portRef A (instanceRef X5))
        (portRef Out (instanceRef X4))
))
(net Net3
    (joined
        (portRef B (instanceRef X5))
        (portRef Out (instanceRef X2))
))
(net A
    (joined
        (portRef A (instanceRef X1))
        (portRef A (instanceRef X2))
        (portRef A)
))
(net B
    (joined
        (portRef B (instanceRef X1))
        (portRef B (instanceRef X2))
        (portRef B)
))
(net S
    (joined
        (portRef Out (instanceRef X3))
```

```
                                        (portRef S)
                                    ))
                                    (net Cin
                                        (joined
                                            (portRef A (instanceRef X3))
                                            (portRef A (instanceRef X4))
                                            (portRef Cin)
                                    ))
                            )
                  ))
                )
                (design ROOT
                  (cellRef Adder1Bit
                        (libraryRef Adder1Bit)))
)
```

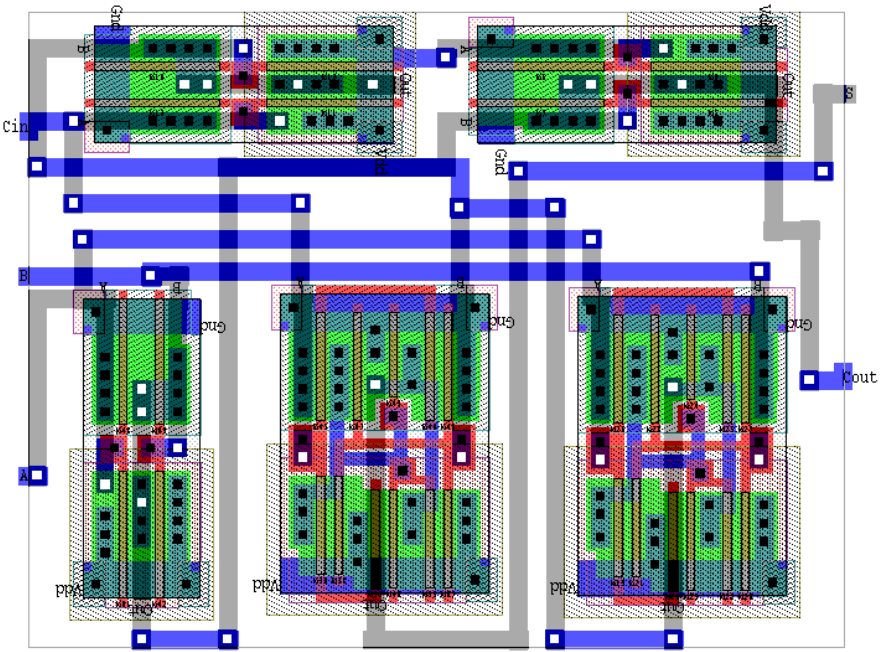
The figure below shows the top-level I/O cell for the same 1-bit adder block. This top-level I/O cell provides BPR with the locations of the external pins to use for routing and indicates the minimum bounding box within which placement and routing will be confined.



The following figure shows the 1-bit adder block placed and routed, without routing to a top-level I/O cell.



The following figure shows the same 1-bit adder block placed and routed, with routing to the external block pins.



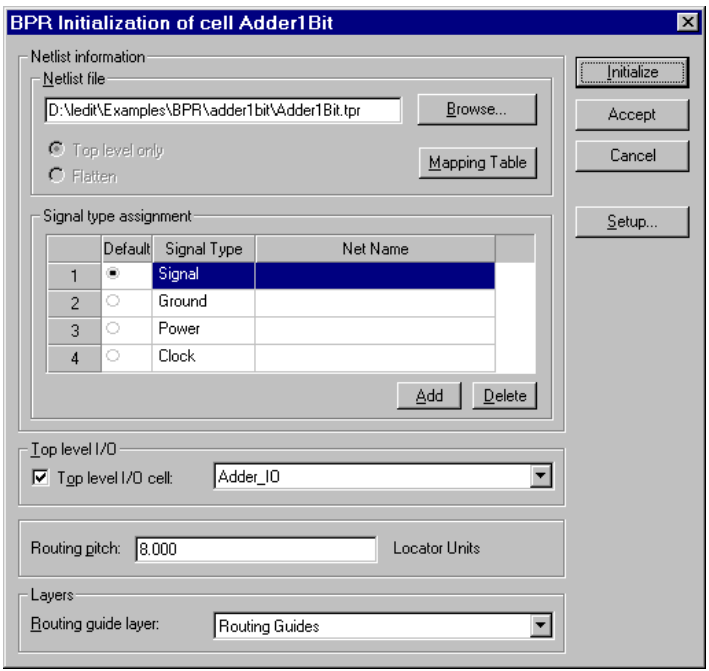
Initialization

During initialization, BPR reads a netlist file which specifies references to cells and connectivity, then places instances of the referenced cells into the design layout with their connectivity shown as routing guides (see Routing Guides on page 2-188). BPR can read netlist files in TPR format or EDIF format (**.edf**, **.edn**, or **.edi**).

The cell that is active when you invoke initialization will be the top-level BPR cell, inside of which all blocks will be placed. If you elect to use a top-level I/O cell, it will contain all referenced blocks, with routing guides connecting external block ports to the I/O pads.

Tools > BPR > Initialization

You initialize a BPR design by opening a design file, opening a cell, and selecting **Tools > BPR > Initialization**. L-Edit displays the following dialog:



Specify initialization options in the following fields:

Netlist file — Designates a TPR or EDIF format netlist file containing cell instance and connectivity information. If your netlist has different cell and port names than those in the current design file, you must map these names using the **Mapping Table**. (See Mapping Table on page 2-180.)

Top-level only — Sets initialization to read the netlist one level down from the top level. (See Top-Level Only/Flatten Option on page 2-178.) This option is available for EDIF format netlists only.

Flatten — Sets initialization to flatten the netlist. See Top-Level Only/Flatten Option on page 2-178. This option is available for EDIF format netlists only; TPR netlists are already flat.

Mapping Table — Opens the mapping dialog for a netlist, which allows you to change cell names and pin names in the netlist to match cell names and pin names in the current design file. For additional information, see Mapping Table on page 2-180.

Default

Sets the default signal type. Any nets in the design not assigned to another signal type during initialization will have this value. Only one signal type can be assigned as the default. The default signal type cannot be deleted.

Signal Type

Designates the signal types used to categorize signals in BPR. You can define your own signal types in addition to the predefined values **Power**, **Ground**, **Signal**, and **Clock**. One signal type must be set as the default.

Click **Add** to add a signal type to the table. Click **Delete** to delete a signal type from the table. Double-click in this field to enter or change a value.

See Signal Types on page 2-233 for more information.

Net Name

Assigns specific nets to a signal type. To assign more than one net to a signal type, separate the net names with a comma. Double-click in this field to enter or change a value.

Top-level I/O cell

Name of the cell that contains ports with the location of the I/O connections that are to be external to the top-level I/O cell. If no top-level I/O cell is specified (when the checkbox is empty), there will be no automatic or assisted manual routing to external pins of the top-level BPR cell.

Routing pitch

Minimum horizontal and vertical distance between the center of two routing wires. This value is used for initial placement to ensure that blocks are placed on the routing grid. If the routing and snap grids are not the same, BPR will prompt you to set your snap grid to match the routing grid pitch.

Routing guide layer

The layer containing routing guides, which show the connectivity between routing ports of nets. Routing guides are used to estimate routing geometry during the placement and routing stages, and to guide you during assisted manual routing.

If you click **Initialize**, L-Edit will confirm that all cells and ports are mapped and, if necessary, will prompt you for corrections before initializing your design.

If you click **Accept**, L-Edit will store your input and close the dialog without confirming any values.

If you click **Setup**, L-Edit will open the **BPR Setup** dialog—see **Tools > BPR > Setup—General** (page 2-201).

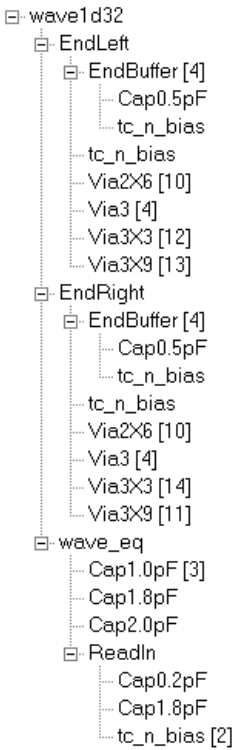
Top-Level Only/Flatten Option

This option is available for EDIF format netlists only. It allows you to either:

- Flatten the netlist, removing the hierarchy
- Use only cells one level down in the hierarchy

In the example shown below, if you specify **Top-level only**, the netlist will consist of three instances: one of **EndLeft**, **EndRight**, and **wave_eq**.

If you specify **Flatten**, the netlist will consists of each instance at the lowest level—**Cap0.5pF**, **tc_n_bias**, **Via2X6 [10]**, **Via3 [4]**, etc.



Mapping Table

The **Mapping: Netlist Cells - TDB Cells** dialog (shown below) allows you to map cell names and pin names in a netlist to their counterparts in your current design file.

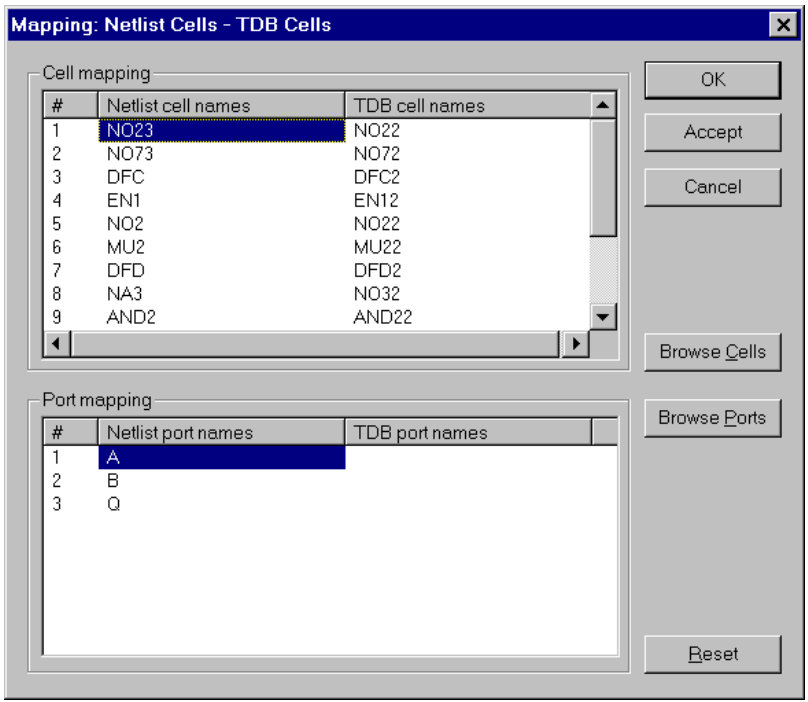
L-Edit invokes this dialog automatically whenever it finds a discrepancy between cell or port names while reading a netlist. You can also open the mapping table directly before initializing BPR by clicking **Mapping Table** in the **BPR Initialization** dialog.

Mapping information is saved in the design file. If you change the netlist (within the same design file), the mapping table will display values for previously mapped cells and ports.

Port mapping is only required if a port name discrepancy occurs. However, if you map one port in an individual cell, you must map all ports for that cell.

If your cell interface in the EDIF file contains ports which are not connected in your design, you can label them as “not used” during the mapping process.

Cell Mapping Dialog



Specify the following options:

Netlist cell names	Shows the names of the cells in the netlist.
TDB cell names	Click in this field to open the **Cells** dialog. Use this dialog to map the highlighted netlist cell to a cell in your design file. You can also use the **Browse Cells** button to open the same dialog.
Netlist port names	Shows the names of the ports in the highlighted netlist cell.
TDB port names	Click in this field to open a **Ports** dialog. Use this dialog to map the highlighted netlist port to a port in your design file.You can also use the **Browse Ports** button to open the same dialog.
OK	Confirms that all cells are mapped, prompts you for corrections or completions if any are necessary, stores your input, and closes the dialog.
Accept	Stores your input and closes the dialog without confirming any values.

Netlist Checking

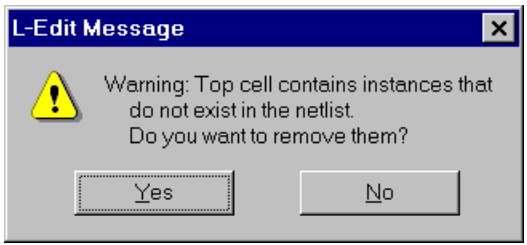
When a netlist is loaded, each instance in the design is checked against the netlist to verify that:

- Instance names match those in the netlist
- Routing port names match the pin names of blocks in the netlist

All instances of the same cell must have a unique instance name both in the netlist and the layout. If the netlist contains non-unique instance names, BPR will not be able to initialize the design.

BPR preserves the placement position and orientation of any cell instances in the design that match instances in the netlist when a netlist is loaded during initialization. If an instance exists in the netlist but not in the design, it will be instanced into the top-level BPR cell.

If BPR finds a cell instanced in the top-level BPR cell that is not in the netlist, it will open a warning dialog prompting you to either delete the instance or keep it in the design.

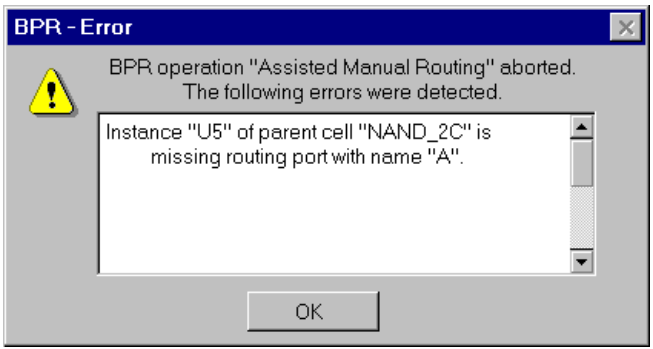


At the start of any BPR operation (placement, assisted manual routing, automatic routing, or editing electrically connected objects), L-Edit will check to make sure the layout is synchronized with the netlist.

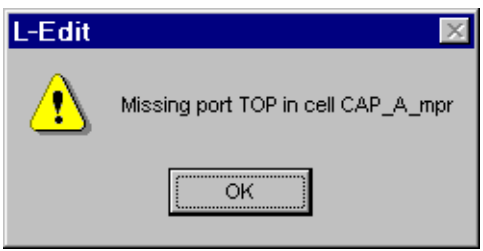
Changes to the layout that affect the netlist include adding or removing instances with connections, adding or removing routing ports, and changing a routing port name.

If the layout is not synchronized with the netlist, L-Edit will display an error message indicating the items that caused the problem and canceling the operation. You should determine the cause of the discrepancy and re-initialize your design with a corrected layout or modified netlist.

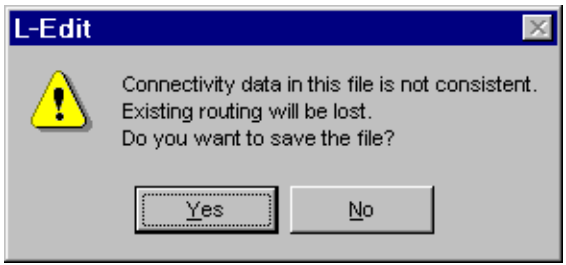
The two figures below show examples of the error messages triggered by a discrepancy between the netlist and the layout. The first shows a missing routing port detected during assisted manual routing.



This message was when a port in the netlist did not appear in the layout.



If you have attempted to save your file and a connectivity check has failed, you will have the option to cancel the save operation.



Re-Initialization

Once a design is initialized and saved, subsequent design stages do not require initialization unless the netlist has changed.

If you change the netlist, you will need to re-initialize your current design. All routing and all layout objects except instances are deleted whenever a design is initialized.

If you like, you can preserve objects on the top-level BPR cell such as keep-outs by grouping them into a cell.

Placement

After a design is initialized, you can arrange blocks in the layout either manually or using the BPR automatic placement feature (see Automatic Placement on page 2-189).

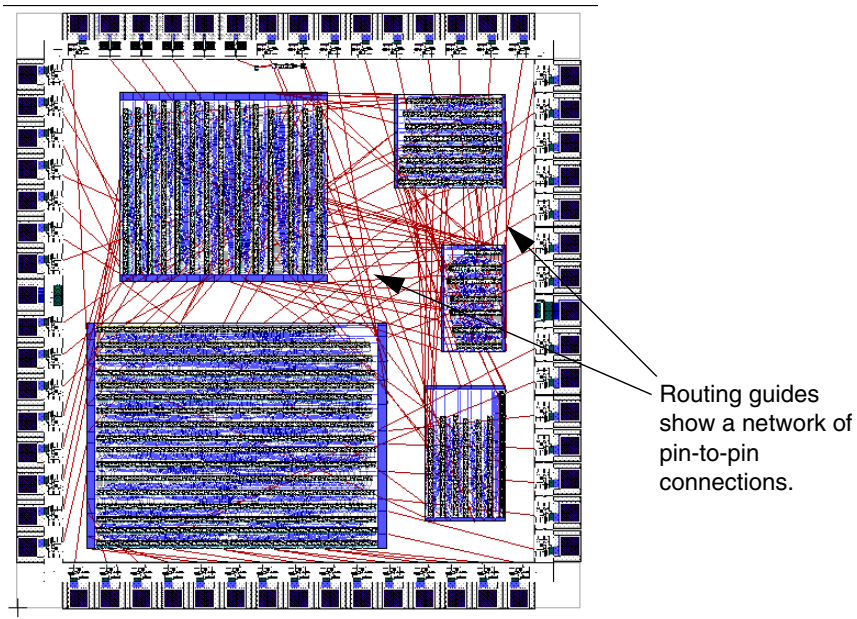
You can change placement of the individual blocks at any point in the design process. You can move any number of blocks at a time. The selected blocks can be manipulated using the **Move**, **Rotate**, **Flip**, **Move By**, and **Nudge** operations. The **Undo** and **Redo** commands are supported for each of these operations.

Routing guides are dynamically updated whenever an unrouted block is repositioned or its orientation is changed. However, if you move a routed block, the existing routing will be deleted and connections will be shown as routing guides. L-Edit will first display a warning dialog allowing you to cancel the operation.

If the design is partially routed, unaffected routing will be preserved when a block is moved.

Routing Guides

A sample ASIC design after initialization and placement is shown below. Connectivity is shown as thin *routing guides* on the routing guide layer.



BPR generates routing guides on the **Routing Guide** layer specified in **Tools > BPR > Initialization** (page 2-174). Lines on this layer cannot be directly edited, but you can modify the layer's visual properties such as color, hatch pattern and line width using **Setup > Layers—Rendering**.

Selecting Routing Guides

Routing guides can be selected by clicking directly on them or by using *edge selection*. To use edge selection, hold the **Ctrl** key while using the left (SELECT) mouse button to draw a box around any portion of the routing guide(s) you want to select. All routing guides within the selection box will be selected. (See also Selecting Nets, Connections, and Objects on page 2-199.)

When you move a block during assisted manual placement, the routing guide display moves along with the block. Similarly, when you draw a wire during assisted manual routing, the routing guide display is updated as your cursor moves to indicate the end point of the connection you are routing.

Edit Object operations cannot be performed on routing guides. If you attempt to edit a set of objects that includes routing guides, the guides will be ignored during the editing operations.

Automatic Placement

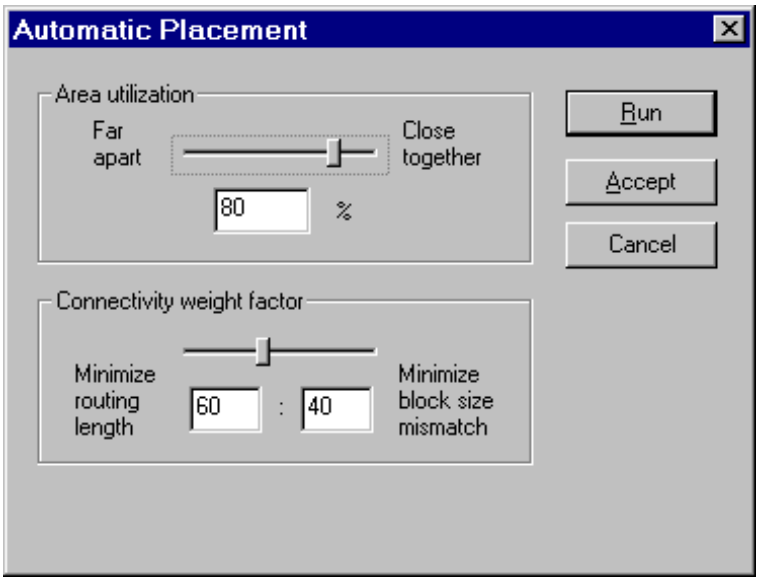
BPR automatic placement uses two compound qualities to place blocks according to the relative importance you have assigned to each.

- **Area utilization** controls how closely together blocks will be spaced. A high area utilization percentage yields closely spaced blocks with little distance between the blocks. A low area utilization percentage yields loosely spaced blocks with a large distance between the blocks.
- **Connectivity weight factor** controls the relative importance of minimizing the routing length or the area wasted due to block size mismatch. BPR computes net lengths based on their topology to determine the routing length and assesses the size, shape, and orientation of each block to determine how they can best fit together to yield the least wasted space in the layout. It then places the blocks according to the relative importance you have assigned to the routing length versus the area wasted when block alignment is not optimized.

If the constraints are conflicting, area utilization will have priority over the connectivity weight factor. For example, if the area utilization value is set very low (so that blocks are placed far apart), then assigning a low relative importance to minimizing the block size mismatch might not have much effect.

Tools > BPR > Automatic Placement

Use the **Automatic Placement** dialog to set automatic placement priorities.



Area utilization

Controls spacing between blocks. A high percentage yields more closely spaced blocks that fill their perimeter geometry more completely. A low percentage yields loosely spaced blocks that do not fill the total available geometry.

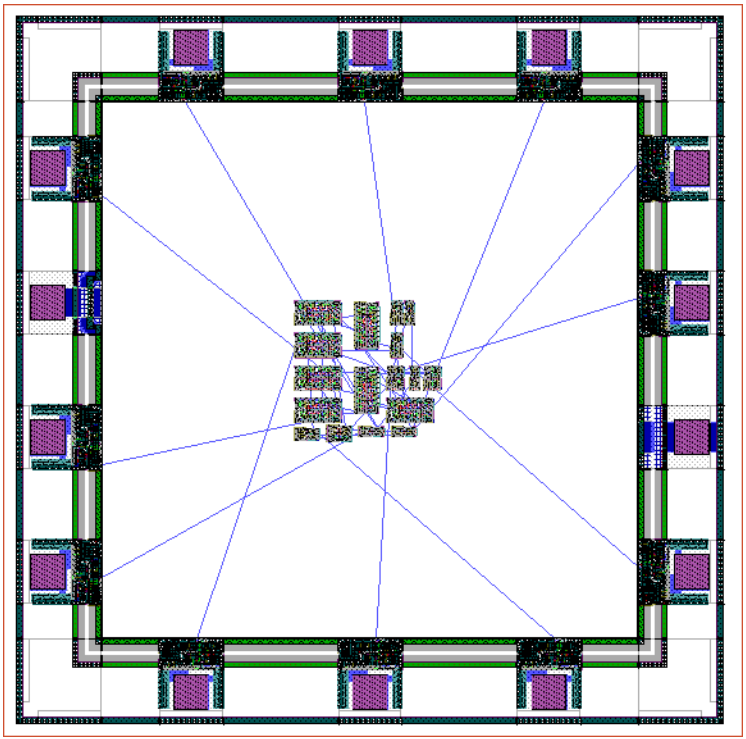
Use the sliding control to set a value from 0% to 100% in 5% increments, or enter any integer from 0 to 100 in the entry field.

If the area utilization setting will place blocks outside the top-level I/O cell, L-Edit issues a warning prompting you to either cancel or proceed.

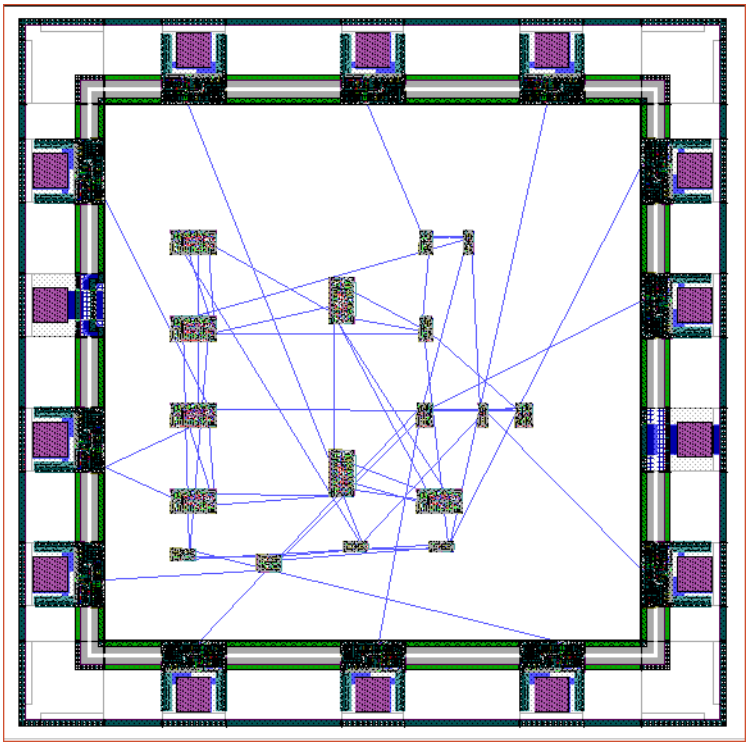
Connectivity weight factor

Sets the relative importance of minimizing routing length or minimizing the area wasted due to block size mismatch. Use the sliding control to set a ratio, or type an integer from 0 to 100 in either of the ratio fields.

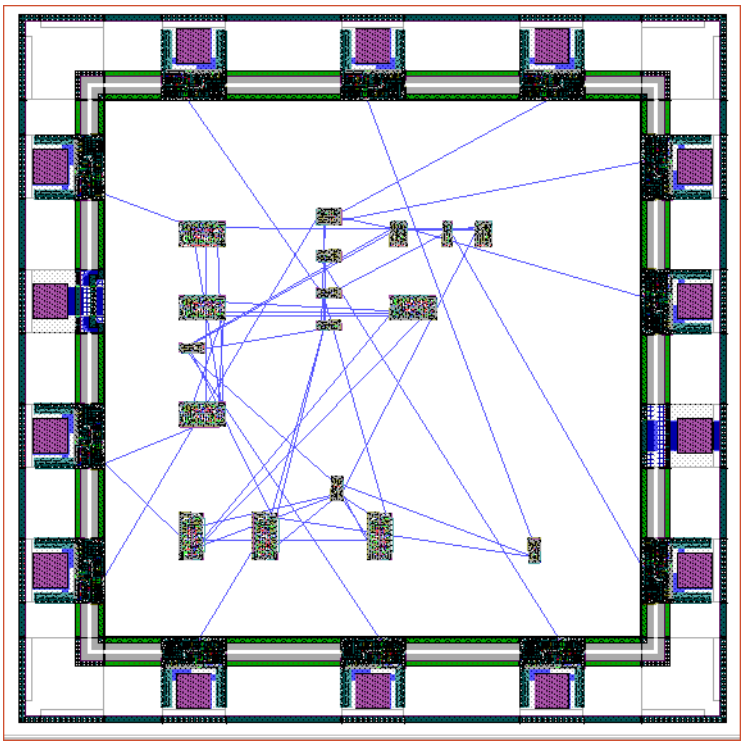
Example of a high percentage area utilization.



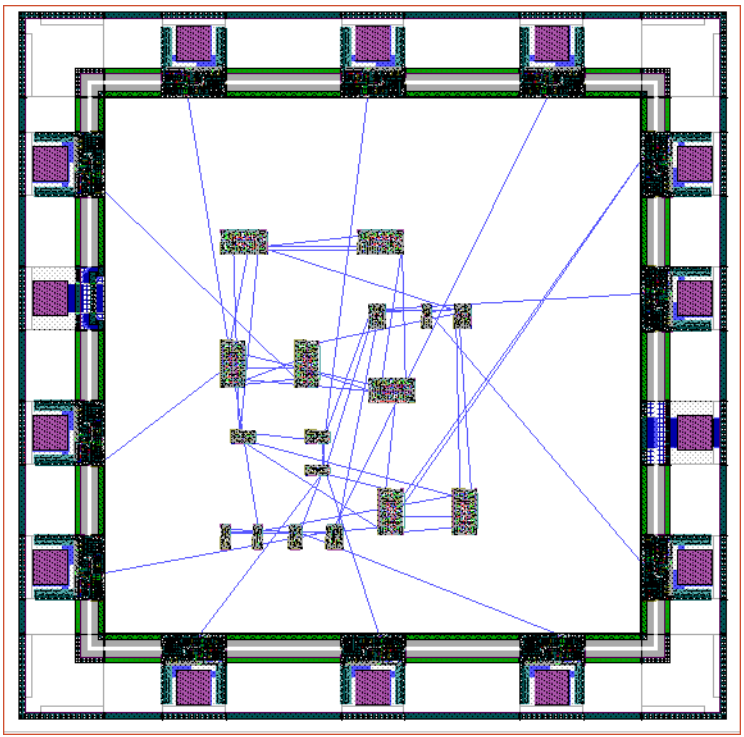
Example of a low percentage area utilization.



Example of minimized routing length.



Example of minimized block size mismatch.



Routing

BPR routing can be performed either automatically or using assisted manual routing to create connections between blocks in a design and, optionally, to a top-level I/O cell.

BPR uses a gridded area router based on V4 routing technology described in Khoo and Cong (see References on page 2-238). Routing is performed on one pair of horizontal and vertical layers at a time.

Some nets are critical to an ASIC design and should be treated separately from the others to satisfy specific performance requirements. Critical nets can be routed first or early in the design process to benefit from the maximum availability of routing resources.

Automatic routing (see Automatic Routing on page 2-204) creates routing wires and vias for all nets in a design. It generates orthogonal routing according to the topologies assigned to each net. Net topology is set in the **Netlist Navigator** (see Routing Topologies on page 2-234). To set the routing layer pair, via cell, keep-out layer, and signal types that will be used during automatic routing, use **Tools > BPR > Setup—Autorouter** (page 2-206).

For the automatic router to run, all required layers and setup information must be specified in the design, and the design file must include all of the blocks and cells specified in the netlist.

Note: BPR automatically changes wire end styles based on the wire join style to avoid discontinuity at wire ends and to make joints compatible. If the **Join style** (in the **Setup Layers**) dialog is **Layout**, **Bevel** or **Miter**, BPR will use the **Extend** wire end style. If the join type is **Round**, BPR will use the **Round** end style. The **Butt** wire end style will not be used in any case.

Assisted manual routing (see Assisted Manual Routing on page 2-215) helps you route manually by attaching a routing guide to your cursor to show you pin-to-pin connections. During assisted manual routing, BPR creates a via whenever you change from one routing layer to another. L-Edit will maintain connectivity if you edit routing by moving a wire or via. Use **Tools > BPR > Setup—Manual Router** (page 2-217) to control the routing layers and vias used during manual routing.

BPR provides orthogonal, 45 degree, and all-angle routing wire drawing tools for assisted manual routing.

Routing Setup

Routing setup consists of the **BPR Setup** dialog, which has three tabs: **General**, **Manual Router**, and **Autorouter**. You must set the appropriate options before you can route your design.

Changes to one tab do not affect the other tab unless you explicitly request that values from the **Autorouter** tab be used to update the **Manual Router** settings.

BPR Setup dialogs are available before and after initialization. Settings are not required prior to initialization, but may be entered at that point. In a shared design environment, for example, a CAD manager might enter L-Edit and BPR setup values on several TBD files before designers run BPR.

Selecting Nets, Connections, and Objects

You can select nets graphically by clicking on them in the layout, or by name in the **Netlist Navigator** (page 2-227).

You can also press **Shift**+SELECT to switch to EXTEND SELECT, where each object selected from the layout is added to the group of objects that are already selected.

The **select nets when selecting instances** option in the **BPR Setup** dialog allows you to toggle whether selecting an instance will also select all nets (or connections) connected to that instance.

Highlighting Selected Nets or Connections

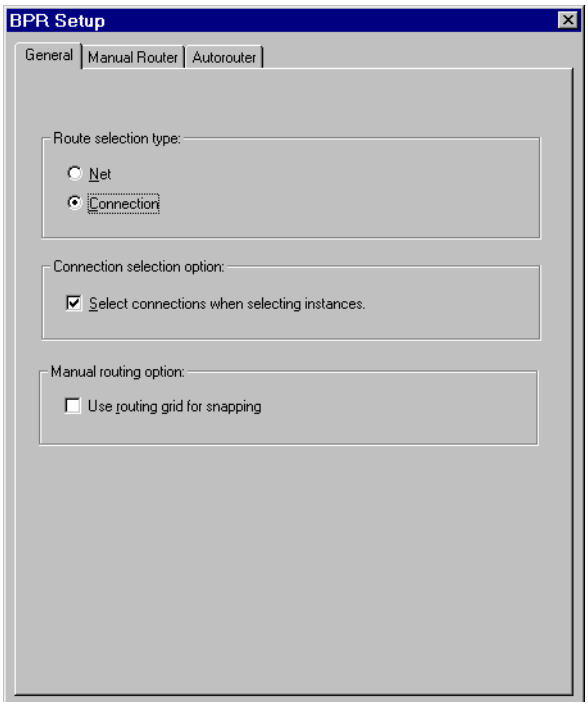
Use Highlight mode ( on the Drawing toolbar) to display the connection or net that a selected object belongs to. Highlight mode toggles only the display of the connection or net that is currently selected—it does not change which objects are selected.

If connection selection is active, just the selected connection will be highlighted. If net selection is active, the entire net that a selected connection or object belongs to will be highlighted.

Clicking any of the other tools on the Drawing toolbar (for example, ) will return you to drawing mode. (The right pane of the status bar indicates the current L-Edit mode.)

Tools > BPR > Setup—General

Use this dialog to set global selection options, and to specify which grid will be used as the snapping grid during manual routing.



Net

When **Net** is checked, clicking on a routing wire or via in the layout will select the entire net that the wire or via is part of for routing, unrouting, timing analysis or signal integrity.

If the **Select nets when selecting instances** checkbox is checked, clicking on an instance will select all nets connected to that instance.

You can also set this option with the **Select Nets/Connections** button () on the Place and Route toolbar.

Connection

When **Connection** is checked, clicking on a routing wire or via in the layout will select just that pin-to-pin connection for routing and unrouting. However, the whole net the connection belongs to will be processed during signal integrity and timing analysis.

If the **Select nets when selecting instances** checkbox is checked, clicking on an instance will select all connections connected to that instance.

You can also set this option with the **Select Nets/Connections** button () on the Place and Route toolbar.

Select nets when selecting instances

When this option is checked, selecting an instance also selects all nets or connections connected to that instance.

Note that you must be in **highlight mode** for the nets associated with the object to *appear* selected in the layout.

During assisted manual routing, it is safer to turn this option off, because any nets or connections explicitly selected will be deselected when you select an instance.

Use routing grid for snapping

When this button is depressed, manual routing wires will snap to the routing grid instead of to the snapping grid intended for other drawing in the layout.// gives you the option to use the routing grid as a snapping grid.

You can also set this option with the **Use routing grid** button () on the Place and Route toolbar.

Automatic Routing

Automatic routing generates orthogonal routing wires and vias between the blocks in the netlist, and to the routing ports of the top-level I/O cell, if one is used.

BPR uses a gridded router that places the center and end-points of routing wires on grid. When a port is not on the routing grid, the router will create a small routing wire segment from the port to the on-grid routing wire to complete wiring.

Keep-out or SRP (subcircuit recognition polygon) boxes prevent routing on both routing layers. The router will allow the center of a routing wire to be located on the boundary of a keep-out region. Routing ports can be located inside a keep-out area within one grid point of the keep-out perimeter.

The **Tools > BPR—Route All** command automatically routes an entire design and **Tools > BPR—Unroute All** automatically unroutes an entire design. You can also automatically route or unroute selected nets or their connections using **Tools > BPR—Route Selected** and **Tools > BPR—Unroute Selected**.

BPR will automatically route the net or connection based on whether net selection or connection selection is active.

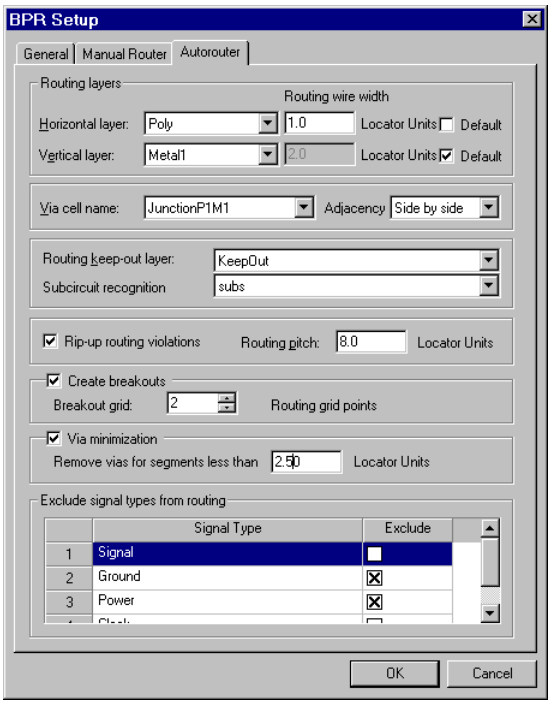
Prior to automatic routing, you must choose one layer for horizontal routing and one layer for vertical routing and set the wire width for each. You also need to set

a routing pitch, choose a via to connect the two routing layers, and specify how closely vias can be positioned. You can also pick a routing keep-out layer if you use one in the design. These choices are all made using **Tools > BPR > Setup—Autorouter** (page 2-206).

Note: Automatic routing eradicates any existing routing for a connection—you cannot partially route a connection manually and then finish the route automatically.

Tools > BPR > Setup—Autorouter

The **Autorouter** dialog specifies options used in automatic routing:



These settings can be completed at any time prior to routing, and you may return to this dialog any time you want to change any of the BPR router settings.

Horizontal layer

Layer on which horizontal signals are routed. This field opens a drop-down list of all layers in the current file.

Vertical layer

Layer on which vertical signals are routed. This field opens a drop-down list of all layers in the current file.

Routing wire width

Specifies the width of the routing wires created on the selected layer. If you check the **Default** check box, the router will use the default wire width from the **Setup Layers** dialog for the selected layer.

Via cell name

Name of the via cell that connects the horizontal and vertical routing layers. This field opens a drop-down list of all cells in the current file.

Adjacency

Specifies the closest allowed spacing for a pair of vias added to a design by the autorouter, to adjust for DRC constraints (see Checking Design Rules on page 3-10). Options are **Side by side** (default) or on the **Diagonal**.

Vias are placed on center on the routing grid. Side-by-side spacing yields the closest via proximity.

Routing keep-out layer

Layer on which keep-outs can be created. This field opens a drop-down list of all layers in the current file.

Subcircuit recognition layer

Subcircuit recognition layer which will be used as an implicit keep-out. This field opens a drop-down list of all layers in the current file.

Note that a value will be imported into this field from the **Subcircuit recognition** field of the **Extract—Subcircuit** dialog. Similarly, an entry in this field will overwrite the **Extract—Subcircuit** dialog value.

Rip-up routing violations

Check this box to remove routing wires that create violations following automatic routing. Wires that are in violation will be replaced with routing guides.

Routing pitch

Minimum allowable distance (horizontal and vertical) between the center of two routing wires.

Create breakouts

Check this box to allow the autorouter to create routing breakouts, small routing segments that are used to reduce routing congestion. (See Breakouts on page 2-212.)

Breakout grid

When breakouts are used, specify an integer value by which the routing grid will be multiplied. Breakouts will be extended to the number of routing grid points entered here. (See Breakouts on page 2-212.)

Via minimization

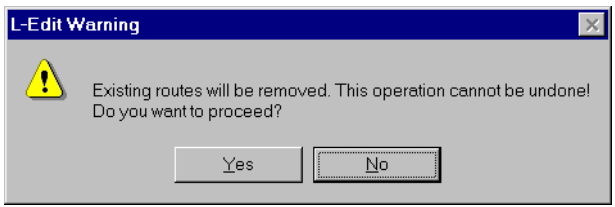
Check this box to remove vias from routing wires that are shorter than or equal to the length specified in the **Remove vias for segments less than** field. Enable this option if the autorouter creates unnecessary routing layer changes.

Exclude signal types from routing

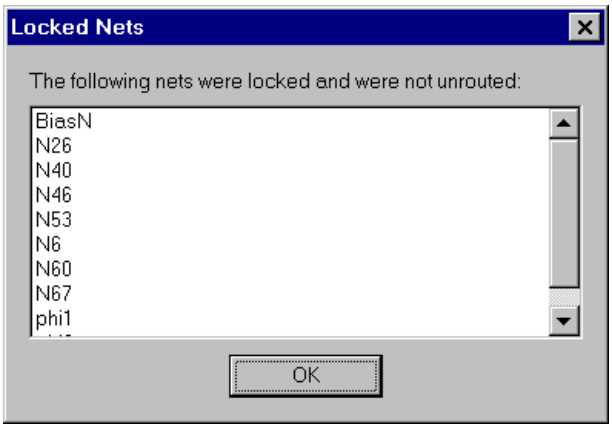
When this box is checked, the selected signal type will be excluded from autorouter processing. The **Signal Type** field contains read-only values.

Preserving Existing Routing

If one or more of the nets selected for automatic routing are already routed, BPR gives you the choice of removing (**Yes**) or preserving (**No**) the existing routes (see below).



The automatic router also preserves any routing that is locked. Use the **Netlist Navigator** to lock or unlock nets. After automatic routing, BPR will display a list of the nets that were locked and therefore not routed or unrouted.



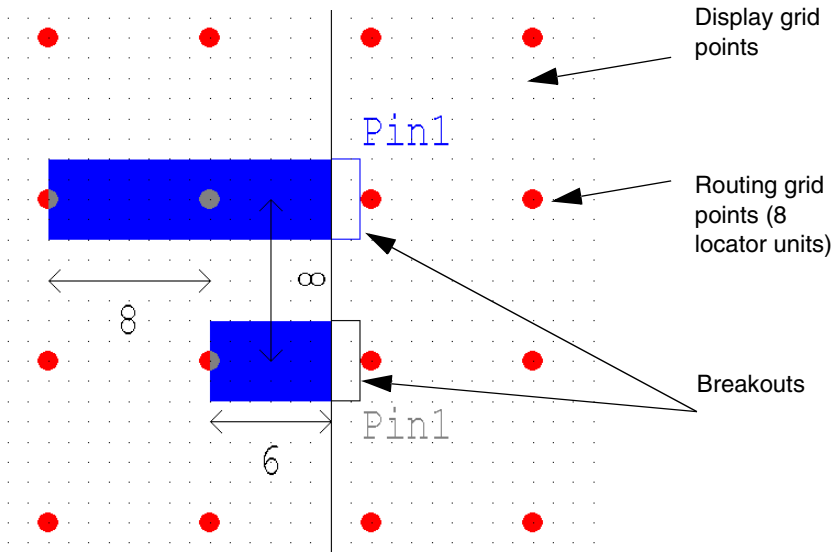
Breakouts

A breakout is a small routing wire that the BPR autorouter extends from a routing port. The autorouter creates breakouts when the number of ports on an edge causes the ports to be too close together for proper routing. Breakouts help the automatic router achieve a higher routing completion rate.

The **Create breakouts** group in **BPR > Setup—Autorouter** allows you to enable or disable breakout creation and to set the maximum number of routing grid points away from a pin that the breakouts can extend.

The maximum allowable breakout extension is calculated as an integer multiple of your routing grid. Breakouts will extend from the port to a point defined by the nearest integer multiple of the routing grid. This multiplier value is specified in the **Breakout grid** field.

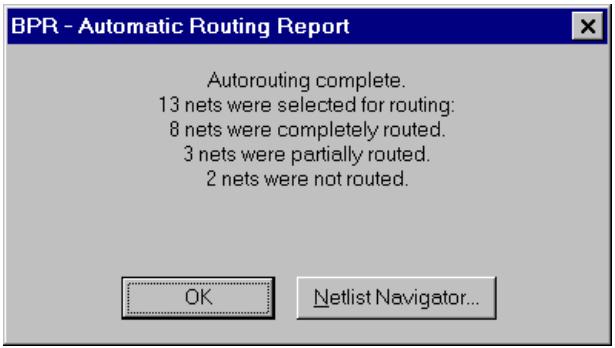
In the figure below, the top breakout was created using a **Breakout grid** value of two routing grid points, while the bottom breakout was created using a **Breakout grid** value of one routing grid point.



Breakout grid points are defined by an integer multiple of the routing grid (red points). Breakouts are extended to points on the breakout grid.

Automatic Routing Report

At the end of automatic routing, BPR displays a summary dialog indicating the number of nets successfully routed (see below).



You can use this routing report to open the **Netlist Navigator** for more comprehensive routing statistics

Multiple Layer Routing

BPR allows you to define as many layers as needed for routing a design. You can route on any number of layers during assisted manual routing. However, the automatic router processes only two layers at a time. To automatically route a design with multiple routing layers, use successive passes of the router and select a different pair of routing layers each time.

You can specify a pair of layers and route some nets, then specify another pair of routing layers and route the remaining nets. You can also route some connections using one pair of layers and then route the rest of the connections using a different pair of layers.

Assisted Manual Routing

Assisted manual routing helps you create or modify routing wires for unrouted or partially routed nets while preserving the net's interconnection topology.

You can add, move, or delete routing wires during assisted manual routing. All connectivity information is preserved and properly updated during the assisted manual routing process.

As you route, the routing guide display attaches to your cursor to show the remaining connection to the receiving pin. This continual updating guides you to the connection points of the net you are routing. BPR also creates a via at each routing layer transition.

When you move a via or BPR routing wire, BPR will automatically update (stretch or collapse) the neighboring connections to preserve the connectivity of the net.

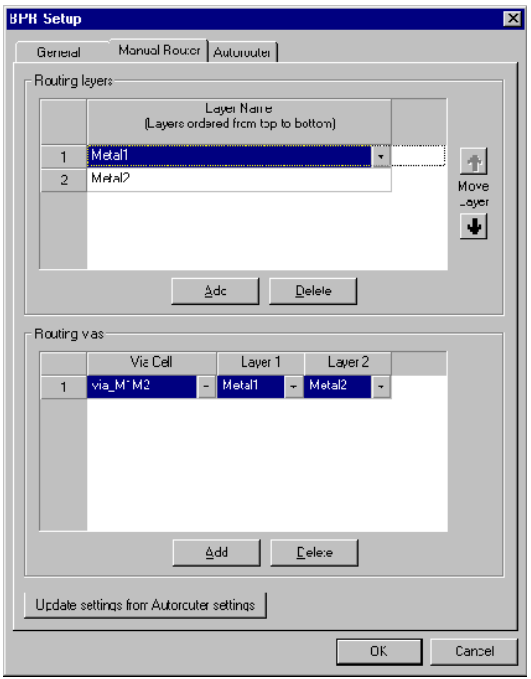
To begin assisted manual routing, select one of the three assisted manual routing wire tools from the Drawing toolbar. If a routing layer is not already active, BPR automatically selects the first routing layer defined in **BPR Setup—Manual Router** as the active layer.

Undo and **Redo** are supported during assisted manual routing. The following restrictions apply to assisted manual routing:

- You can only route on layers defined as manual routing layers in **Tools > BPR > Setup—Manual Router** (page 2-217).
- You must route existing unrouted connections; you cannot create new connections.
- Routing must begin at the end point of a routing guide.
- Keep-out are ignored.

Tools > BPR > Setup—Manual Router

The **Manual Router** dialog specifies the routing layers and vias used during manual routing.



These settings can be completed at any time prior to routing.

Layer Name

Layer on which BPR will route nets. This field opens a drop-down list of all layers in the current file. Default is the current layer.

Move Layer

Arrow buttons used to move the selected layer(s) up or down in the routing order. The layer order in this list should correspond to the actual fabrication order of the design.

Add

In the **Routing layers** group, adds a new layer to the **Layer Name** list. In the **Routing Vias** group, adds a new row to the table.

Delete

In the **Routing layers** group, deletes the selected layer from the **Layer Name** list. In the **Routing Vias** group, deletes the selected via from the table.

Via Cell

Cell used to create a routing via between the two specified layers. This field will present a drop-down list of all cells in the current file.

Layer 1/Layer 2

Layers connected by the selected via. This field opens a drop-down list of all manual routing layers in the current file.

Update settings from autorouter settings	Copies the layer and via definitions from the **Autorouter** tab to the **Manual Router** tab.

Cycling Through Routing Layers

L-Edit provides menu and keyboard shortcuts that allow you to cycle through the defined manual routing layers as you route with the assisted manual router.

Use **Draw > Next Routing Layer (F11)** to cycle down and **Draw > Previous Routing Layer (F12)** to cycle up the list of routing layers as they appear in the **Layer Name** list in the **BPR Setup—Manual Router** dialog. Both functions cycle continuously through the layer list.

Next Routing Layer and **Previous Routing Layer** change the layer of the routing wire you are *currently* drawing. This is true whether you draw wires using a "rubberbanding" mouse operation, where you do not hold the mouse button down as you draw, or a "click and hold" mouse operation—the layer will change for the wire that has not yet been completed.

For example, suppose your manual routing layers are **Metal4**, **Metal3**, **Metal2**, and **Metal1** (listed in that order in the **Layer Name** list of the **BPR Setup—Manual Router** dialog), and you have one via defined for each possible layer connection.

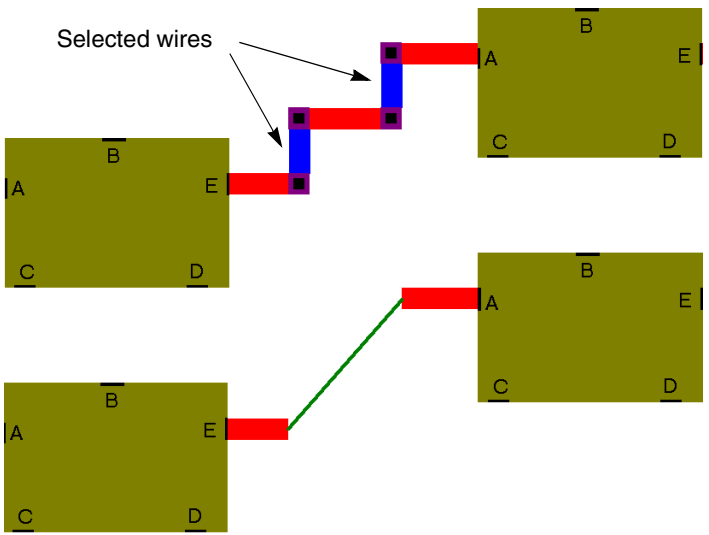
If you begin assisted manual routing on **Metal2** and you click with the left mouse button (DRAW) to begin a wire, pressing **F12** will cycle up the list to change the layer on which that wire is drawn to **Metal3**. If you click the right mouse button (VERTEX) to begin a second wire and press **F12** again, BPR will cycle up the list so that the second wire is drawn on **Metal4** and an instance of the via that connects **Metal3** and **Metal4** is added.

If you have defined a via that connects the layer you are switching from and the layer you are switching to, BPR will add that via at the layer change. If you have not defined such a via, BPR will display the error message shown below.

Undo does not affect layer changes. It will revoke the via or wire that was last added to the layout. Similarly, **Redo** will reinstate the last object changed by **Undo**.

Deleting Multiple Routing Wires

If multiple selected routing wires are not consecutive, **Cut** and **Delete** will delete the selected wires and any routing between them, as shown below:



When two disjoint wires are selected and deleted, any wires between them will also be deleted.

Operations on Electrically Connected Objects

Electrically connected BPR objects cannot always be edited in the same way that their non-electrically connected counterparts can be. For instance, routing wires are subject to editing constraints not placed on other wires. Similarly, vias and BPR blocks cannot be manipulated as freely as cell instances.

You can use **Copy** and **Duplicate** to convert an electrically connected object to its non-electrically connected counterpart when it is placed in the internal buffer. **Paste** will paste the non-electrically connected object. For example, if you **Copy** a block, **Paste** will convert it to an instance. If you **Copy** a route, **Paste** will convert it to wires and via instances. To convert a top-level BPR cell to a cell without BPR connectivity, use **Cell > Copy**.

If you initiate a disallowed operation on a selection set that includes electrically connected objects, the operation will be completed for all other objects but ignored for the electrically connected objects.

The table below shows which operations are disallowed for specific electrically connected objects, and any limitations.

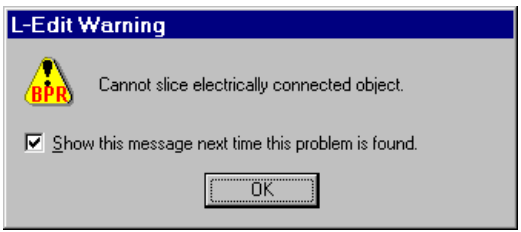
Object	*Disallowed Operations*	*Allowed Operations*
Box, Polygon, Wire, Circle, Port, Ruler, Instance	None	All
Block	Cut Delete Edit-in-Place Group/Ungroup	Copy Flip Rotate Move By Move/Edit Nudge

Object	*Disallowed Operations*	*Allowed Operations*
Routing Wire	Flip Group/Ungroup Merge Nibble Rotate Slice	Copy Cut Delete (When you delete a routing wire it will be replaced with a routing guide.) Move/Edit Move By Nudge Textual Editing (Move By, Move/Edit and Nudge are allowed only if one wire is selected, otherwise the operations will be ignored.)

Object	*Disallowed Operations*	*Allowed Operations*
Via	Flip Group/Ungroup Rotate	Copy Cut Delete Move By Move/Edit Nudge Textual editing (Cut and delete are allowed if the via is connected to a routing guide but not if it is connected with two wires.)

Disabling Warnings

A warning about a disallowed BPR operation can be disabled, either in the warning dialog itself (as shown in the figure below), or using **Setup > Application—Warnings**.

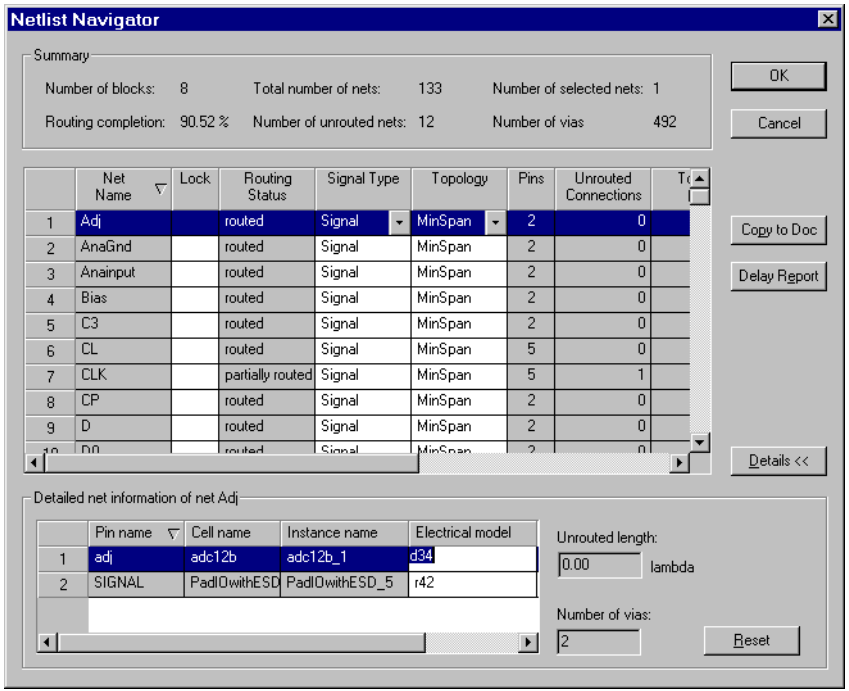


Netlist Navigator

The **Netlist Navigator** is a netlist browser that allows you to:

- View net information—the number of pins, number of unrouted connections, total net length, maximum delay, and skew calculated for the net
- Select, sort, and change values for one or more nets
- Hide the routing guide display for a net
- View and edit a net's signal type
- View and edit the routing topology for each net (See Routing Topologies on page 2-234)
- Lock a net to prevent any changes to the existing routing
- View the instance name and pin definition associated with a selected pin, and view and edit the electrical model name.
- View the number of vias and the unrouted length for a selected net

Tools > BPR > Netlist Navigator



The **Summary** group shows the following information for the entire design:

Number of blocks	Total number of blocks in the design.
Routing completion	Percentage of nets that are completely routed.
Total number of nets	Total number of nets in the design.
Total number of unrouted nets	Total number of unrouted nets in the design.
Number of selected nets	Number of nets currently selected.
Number of vias	Total number vias in the design.

The unlabeled middle group shows the following information:

Net Name	Name of the net. Values in this column are sorted alphabetically and then numerically (for example, N1, N2, N10, N20).
Lock	Indicates whether a net is locked or not. Double-click to turn net locking on and off. A locked net cannot be routed, unrouted, edited, or deleted. Instances connected by locked nets cannot be moved or deleted.

Routing Status	Indicates whether the net is completely routed, partially routed, or unrouted. This field is read-only.
Signal Type	Indicates a net's signal type. (See Signal Types on page 2-233.)
Topology	The topology used to route a net. Minimum spanning tree topology is the default used for routing guides during initialization. (See Routing Topologies on page 2-234 for more details.)
Pins	The number of pins associated with a net. This field is read-only.
Unrouted Connections	The number of unrouted pin-to-pin connections. This field is read-only.
Total Length (Lambda)	The sum of the lengths of all routed and unrouted connections. This field is read-only.
Hide Guides	Indicates whether routing guides for the net are shown or hidden in the layout.
Max Delay (ns)	Maximum delay from the source pin to all target pins, in nanoseconds. This field is read-only.

Skew (ns) — Maximum delay minus the minimum delay from the source pin to all target pins, in nanoseconds. This field is read-only.

Click on **Details** to open the **Detailed net information** group at the bottom of the **Netlist Navigator**. This group allows you to edit the pin type and electrical model definitions for a single net, and displays routing wire length information. (Values are displayed only when just one net is selected.)

Pin name — Name of the port associated with the pin. This field is read-only.

Cell name — Name of the cell the pin belongs to. This field is read-only.

Instance name — Name of the instance the pin belongs to. This field is read-only.

Electrical model — Use this field to specify the name of the external subcircuit model in the SPICE netlist that defines the behavior of the driver or receiver associated with the pin.

Source/Target	Currently defined pin type, either **Load**, **Source**, or **Target**. (See Routing Topologies on page 2-234 for more details.) If the net is routed, a change to the pin type may cause BPR to unroute the net, depending on the topology.
Unrouted length	The sum of the lengths of all unrouted connections.
Number of vias	The number of vias for the selected net.
Reset	Returns all pin values to what they were when the group first opened.
Copy to Doc	Copies net information (but not pin information) into a text document that is displayed onscreen. Each table row is written to one line in the text file, and table field values are separated with a tab.
Delay Report	Opens a **Delay Calculation Report** for the selected nets. Refer to the section Delay Calculation Report on page 2-248 of the *Timing Analysis and Signal Integrity in BPR* chapter for a complete description of this feature.

Signal Types

Signal types are used to categorize nets so they can be easily sorted and selected in the **Netlist Navigator**. You can change a value for all the selected nets at once. For example, you could group nets by signal type to show or hide their routing guides. You can also exclude nets by signal type from automatic router processing using the **Exclude signal types from routing** group in **BPR Setup—Autorouter**.

Signal type values are defined using **Tools > BPR > Initialization** (page 2-174). The are four predefined values: **power**, **ground**, **signal**, and **clock**. You can delete or change the names of these predefined signal types. You can also define your own signal types. However, changes to signal type don't take effect until a design is initialized.

One signal type must be assigned as the default. During initialization, signal types are compiled from those designated in the **BPR Initialization** dialog and assigned to nets. Any nets not explicitly assigned a signal type will be assigned the default signal type. The default signal type cannot be deleted. You must initialize a design to create or modify a signal type.

Routing Topologies

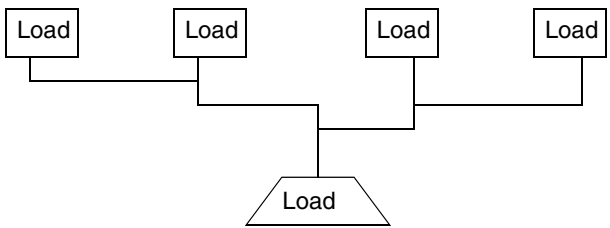
BPR supports the following routing topologies:

- Minimum spanning tree topology
- Star topology
- Daisy chain topology

Minimum spanning tree is the default topology. Some routing topologies require specification of source and target pins for the net. Both the source and the target pin should be defined for daisy chain topology. The source pin should be defined for star topology. If you don't specify the required source or target pins, L-Edit will warn you when you try to close the **Netlist Navigator** and during routing, where no action will be performed.

Minimum Spanning Tree Topology

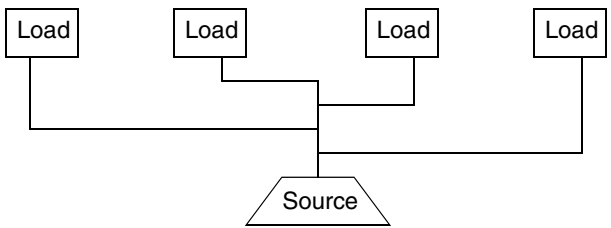
Minimum spanning tree topology creates routing that minimizes the total length of all pin-to-pin connections. Because interconnection length is minimized, it is useful for routing noncritical nets. This topology ignores performance-related issues.



Minimum Spanning Tree Topology

Star Topology

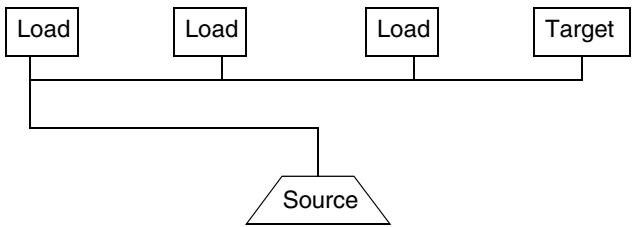
Star topology is helpful when the critical net constraint is maximum delay. This topology is used to minimize signal delay and skew from the given source to any target node of the net. A source pin must be designated for nets using a star topology.



Star Topology

Daisy Chain Topology

Daisy chain topology connects blocks serially from the source pin to a target pin. It is usually used when signal quality is the greatest design concern. Due to the absence of side paths, reflection and ringing are reduced, but this topology can lead to excessive delay. Both a source and target pin must be defined for nets using this topology



Daisy Chain Topology

References

K-Y Khoo and J. Cong. An efficient MCM router based on four-via routing. In *Proceedings of the ACM/IEEE Design Automation Conference*, pp 590-595, June 1993.