

# Overview

---

- [L-Edit: An Integrated Circuit Layout Tool](#)
- [Documentation Conventions](#)



# L-Edit: An Integrated Circuit Layout Tool

L-Edit is a layout tool that represents the masks that are used to fabricate an integrated circuit. In L-Edit, layers are associated with masks used in the fabrication process. Different layers can be conveniently represented by different colors and patterns.

L-Edit describes a layout design in terms of files, cells, instances, and mask primitives. You may load as many files as desired into memory. A file may be composed of any number of cells. A file may be composed of any number of cells. These cells may be hierarchically related, as in a typical design, or they may be independent, as in a “library” file. Cells may contain any number or combination of mask primitives and instances of other cells.

## Cells: The Basic Building Blocks

The basic building block of the integrated circuit design in L-Edit is a cell. Design layout occurs within cells. A cell can:

- Contain part or all of the entire design.
- Be referenced in other cells as a sub-cell, or *instance*.
- Be made up entirely of instances of other cells.
- Contain original drawn objects, or *primitives*.

- Be made up entirely of primitives or a combination of primitives and instances of other cells.

## Hierarchy

L-Edit supports fully hierarchical mask design. Cells may contain instances of other cells. An instance is a reference to a cell; should you edit the instanced cell, the change is reflected in all the instances of that cell. Instances simplify the process of updating a design, and also reduce data storage requirements, because an instance does not need to store all the data within the instanced cell — instead, only a reference to the instanced cell is stored, along with information on the position of the instance and on how the instance may be rotated and mirrored.

There is no preset limit to the size or complexity of the hierarchy. Cells may contain instances of others cells that in turn contain instances of other cells, to an arbitrary number of levels (subject only to hardware constraints).

L-Edit does not use a “separated” hierarchy: instances and primitives may coexist in the same cell at any level in the hierarchy.

Design files are self-contained. The “pointer” to a cell contained in an instance always points to a cell within the same design file. When cells are copied from one file to another, L-Edit automatically copies across any cells that are instanced by the copied cell, to maintain the self-contained nature of the destination file.

## Design Rules

Manufacturing constraints can be defined in L-Edit as design rules. Layouts can be checked against these design rules.

## Design Features

L-Edit is a full-custom mask editor. Manual layout can be accomplished more quickly because of L-Edit's intuitive user interface.

In addition, you can construct special structures to utilize a technology without worrying about problems caused by automatic transformations. Phototransistors, guard bars, vertical and horizontal bipolar transistors, static structures, and Schottky diodes, for example, are as easy to design in CMOS-Bulk technology as are conventional MOS transistors.

## Floorplans

L-Edit is a manual floorplanning tool. You have the choice of displaying instances in outline, identified only by name, or as fully fleshed-out mask geometry.

When you display your design in outline, you can manipulate the arrangement of the cells in your design quickly and easily to achieve the desired floorplan.

You can manipulate instances at any level in the hierarchy, with insides hidden or displayed, using the same graphical move/select operations or rotation/mirror commands that you use on primitive mask geometry.

## File Formats

L-Edit can import and export two standard mask layout interchange formats (GDSII and CIF) as well as read and write Tanner Research's proprietary binary database format, TDB (Tanner Database).

TDB files contain the same information as CIF or GDSII files, plus additional information, such as the last view of each cell, user preferences, and technology information (e.g., layer names, colors, stipple patterns, and grid and design rules).

During the design process, the TDB format is used to load and store design data. The CIF or GDSII formats are typically used to transfer designs to a fabricator or to a CAD platform other than L-Edit.

Although CIF and GDSII are industry-standard formats, each design tool employs its own customizations and extensions and enforces its own limitations within the standard format. L-Edit's interpretations of these formats are fully described in this guide.

## Memory Limits

In L-Edit, you can make your design files as large as you like, given available RAM and disk space.

## Hard Copy

L-Edit provides the capability to print hard copy of the design. A multipage option allows very large plots to be printed to a specific scale on multiple 8 1/2 x 11 inch pages. An L-Edit macro is available to support large-format, high-resolution, color plotting on inkjet plotters.



## Variable Grid



L-Edit's grid options support lambda-based design as well as micron-based and mil-based design.

## Error Recovery

L-Edit's error-trapping mechanism catches system errors and in most cases provides a means to recover without losing or damaging data.

## L-Edit Modules

- L-Edit<sup>TM</sup> : a layout editor
- L-Edit/ SPR<sup>TM</sup> : an automatic standard cell placement and routing package
- L-Edit/ Extract<sup>TM</sup> : a layout extractor
- L-Edit/ DRC<sup>TM</sup> : a design rule checker

L-Edit is a full-featured, high-performance, interactive, graphical mask layout editor. L-Edit generates layouts quickly and easily, supports fully hierarchical designs, and allows an unlimited number of layers, cells, and levels of hierarchy. It includes all major drawing primitives and supports 90°, 45°, and all-angle drawing modes.

L-Edit/ SPR generates layouts for standard cell design and can automatically construct entire chips. It includes cell placement and routing, pad frame generation, and pad routing. SPR accepts netlists produced by S-Edit and creates masks ready for fabrication.

L-Edit/ Extract creates SPICE-compatible circuit netlists from L-Edit layouts. It can recognize active and passive devices, subcircuits, and the most common device parameters, including resistance, capacitance, device length, width, and area, and device source and drain area.

L-Edit/ DRC features user-programmable rules and handles minimum width, exact width, minimum space, minimum surround, non-exist, overlap, and

extension rules. It can handle full chip and region-only DRC. DRC offers Error Browser and Object Browser functions for quickly and easily cycling through rule-checking errors.





# Documentation Conventions

This section contains information about the typographical and stylistic conventions that this guide uses to present information.

*Inline references* to commands, special characters, filenames, and examples of user input and program output are represented by a bold font. An example of a filename is **Layout1.tdb**.

*Elements in hierarchical menu paths* are separated by a > sign. For example, **File > Open** refers to the **Open** command in the **File** menu.



*Freestanding quotations* of input examples, file listings, and output files are represented by a constant-width font.

*Variables* are presented in an italic font.

*Text omitted* for clarity or brevity is indicated by an ellipsis (...).

*Special keys* are represented by abbreviations, as follows.

<i>Key</i>	<i>Abbreviation</i>
Shift	<b>Shift</b>
Return/Enter	<b>Return</b>



<i>Key</i>	<i>Abbreviation</i>
Control	<b>Ctrl</b>
Alt *	<b>Alt</b>
Backspace	<b>Back</b>
Delete	<b>Del</b>
Insert	<b>Ins</b>
Home	<b>Home</b>
End	<b>End</b>
Page Up	<b>PgUp</b>
Page Down	<b>PgDn</b>
Function Keys	<b>F1 F2 F3 ...</b>
Arrow Keys	↓, ←, →, ↑

\* The **Alt** key does not always work in *Unix*. In such cases, use the **Meta** (◆) key instead.

When combinations of keys are to be pressed simultaneously, their abbreviations are adjoined by a plus sign (+); for example, **Ctrl+R** means that the **Ctrl** and **R** keys are pressed at the same time.

When combinations of keys are to be pressed in sequence, their abbreviations are separated by a space ( ); for example, **Alt+E R** means that the **Alt** and **E** keys are pressed at the same time and then released, immediately after which the **R** key is pressed.

Slashes (/) separate alternatives; for example, **Shift+↑/↓** means that the **Shift** key can be pressed together with either the up arrow key or the down arrow key.