# Layout Windows

Layout Windows™

Universal Multiwindow
Layout Editor
version 1.28

User's Manual

KIPARIS Group

Silicon Company, MIEE,
Zelenograd, Russia, 103498
phones 7-095-532-9967
7-095-532-9861
fax 7-095-532-1708
e-mail sokolov@miet.sensor.zgrad.su

Table of Contents

Section 1
Getting Started

Introduction to LW

"Layout Windows" (LW) is a universal program to create and edit VLSI and PCB layouts, schematic and other drafts in CAD systems. The editor supplies flexible multiwindow user interface. DRC rules may be applied to VLSI and PCB files.

Main features

Friendly user interface:

- you can create as many windows as you want - without any limits of shape and placement with pleasant colours

- 3 types of menu commands, submenus, quick jumps between submenus, automatic storing and retrieving menu state

- special graphics representations: "back view", "feet up", 9 auxiliary grids, snapping; independent grid steps forward X and Y co-ordinate axes

- unusual quick drawing, even through the deep hierarchy

- multiply languages of message and menus, opportunity to select acceptable detail level

- easy and direct layer table control, including activity, visibility, draw order and draw patterns

Graphics primitives:

- rectangles, polygons, wires and lines are available

- other file instances with optional rotate, mirror and scale factors

- automatic arcs, rings and circles creation

- angle checks

- any primitives may be multiplicated, inclined arrays are available either

- English and national layout text libraries: you can scale, rotate and mirror text and even create your own text libraries

- CIF and SOURCE file input and output

Edit operations and other features:

- it is possible to copy or move primitives both in one and different windows by easy and clear way

- independent transfer modes: rotation, mirror, scale, corrections of size and layer change

- deformation; wire cut; path change and other

- wide possibilities to select primitives for operations: in active, visible or in all layers; select by point, area or full file space; using marked primitives

- auto or manual electric node setup while routing

- local or full manufacture's rules check by means of menu commands or special file with rules; shorts checking

- simultaneous multiply file processing, automatic library file protection to avoid destroy of data

Software/hardware basis:

- MS-DOS, VMS, Sun/OS and other UNIX-style operating systems, Microsoft Windows and Windows NT

- IBM PC video adapters, Tektronix, HP, DEC and other third-party terminals support

- best choice for X Window System environment and X-terminals

Starting LW

To start LW type in the command line:

> **lw**

and press the ENTER key. In window equipments like OPEN LOOK or Microsoft Windows pick LW's icon in File Manager or use any possible way.

Just picture, shown in Figure 1, appears, you can begin to work with editor.
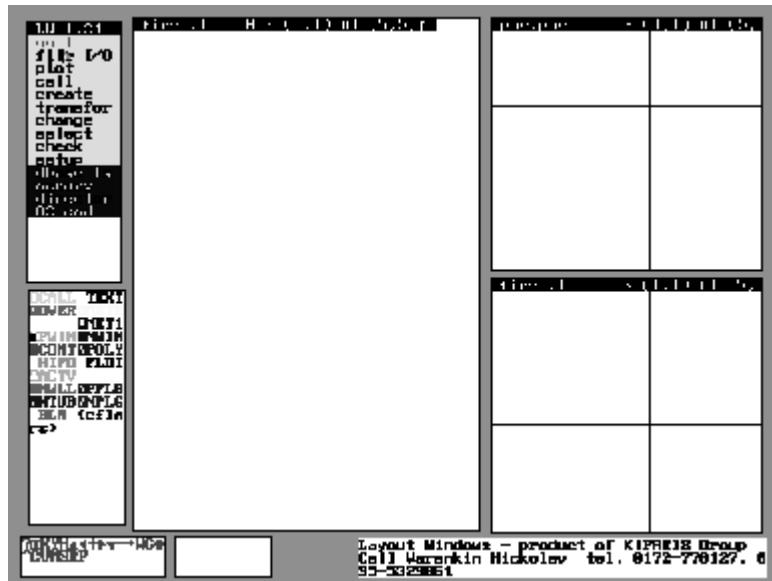
Figure 1 Rough Editor's View

To quit from LW, pick the **"quit"** command in the MAIN menu.
If you have drawn something and did not store it, the
editor asks you:

    Really quit?(y/[n])

To cancel the quit, enter the **"n"** character (or press
ENTER at once). In opposite case enter the **"y"** character
and press the ENTER key. After that the edit session is
finished.

Section 2
LW Concepts

Design objects

The basic structure of a LW design is the <u>cell</u>. Cells consist of objects, named <u>primitives</u>. Primitives can be placed on various layers. The cell can include special objects named <u>instances</u>, in addition to usual primitives like polygon or wire. Instance is a reference to other cell. When you place the instance into the cell, instantiated data are not copied.

Primitives

Primitives are divided in:

- **basic** primitives

- **macro primitives:** primitives, based on basic primitives

The basic primitives in LW are the following:

- **box**, which is specified by its sides. Sides of the box are always parallel to corresponding co-ordinate axes. Figure 2 shows an example of a box you can create in LW.
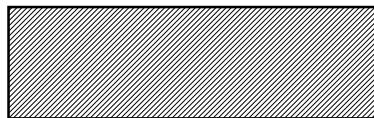
Figure 2 Example of a Box

- **polygon:** object, which is specified by its outline.

    There are two restrictions for polygons:

    - polygon can't contain acute internal angles.

    - polygon can't have more than 60 tops. It's connected with restrictions in SOURCE format.

    Figure 3 shows example of polygon.
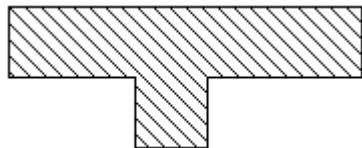
Figure 3 Example of a Polygon

- **wire.** This is object, which is specified by its path, width and end margin. Figure 4 shows a wire, its path (vertices of path marked by crosses), and there is its width and positive end margin.
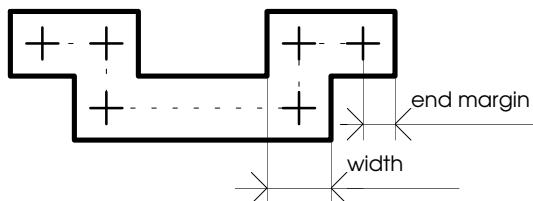
Figure 4 Schematic View of Wire

The wire width must be more than zero. The value of end margin might be positive, negative or equal to zero. In the last case the startpoint and the endpoint of path will be out of wire outline.

There are two restrictions for wires:

- wire path can't contain acute angles.

- wire can't have more than 30 vertices. It's connected with restrictions in SOURCE format.

Figure 5 shows example of wire.
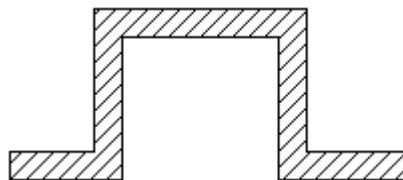
Figure 5 Example of a Wire

- **line.** This is the wire with the zero width. The end margin is not specified.

There is one restriction for lines:

- line can't have more than 60 vertices. It's connected with restrictions in SOURCE format.

Figure 6 shows example of line.
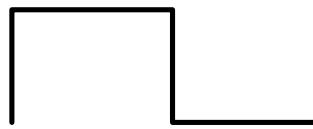
Figure 6 Example of a Line

- **instance.** It's the reference to the cell, which can be placed in the layout of the other cell. The instance is specified by the cell name and the base point. For more information see the chapter *Cells and instances* in this section. Figure 7 shows the instance example.
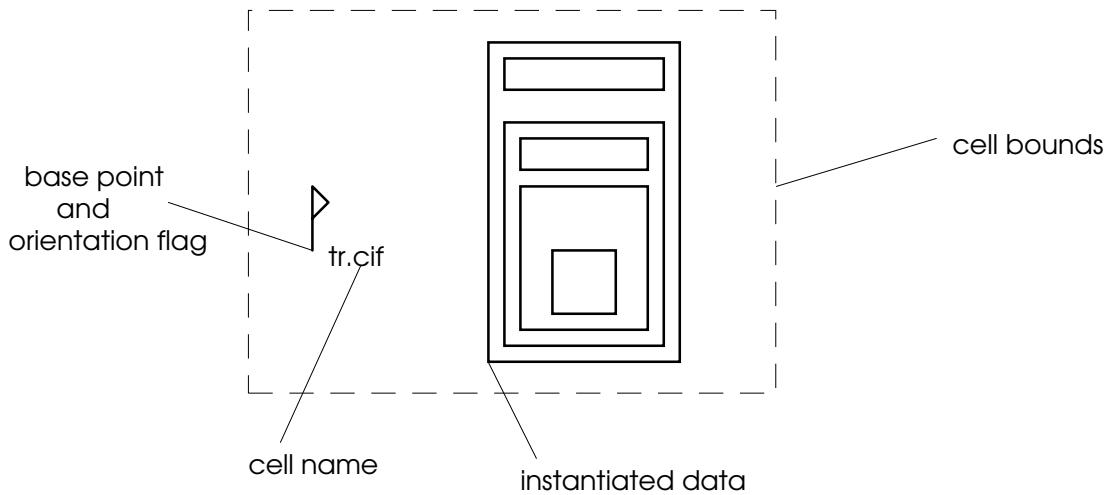
Figure 7 Example of an Instance

You can create an array of any basic primitives. The array is specified by the following parameters:

- directions of the element's spreading

- number of elements in these directions

The directions of the element's spreadings in the array M and N are specified by the vectors (mx, my) and (nx, ny) correspondingly. These vectors also specify the distance between the neighbouring elements of array. In common case the array is oblique-angled. Figure 8 explains the sense of M and N parameters.
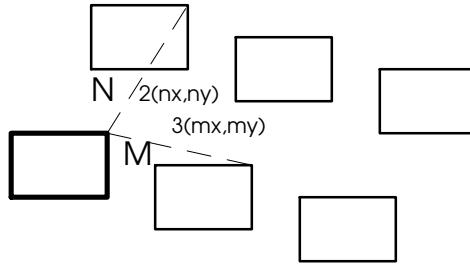
Figure 8 Parameters of Array

You can specify rectangular array, if the vectors M and N are set as follows:

M (mx, 0)   N (0, ny),

where   mx is horizontal distance between neighbouring elements,

     nx is vertical distance between neighbouring elements.

In the Figure 9 you can see the rectangular array.
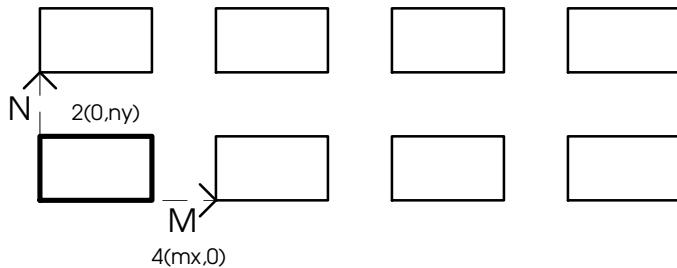
Figure 9 Example of Rectangular Array

Macro primitives are represented inside the editor by the primitives with the definite properties. They are:

- **circle.** It's the object, which is specified by its centre and radius. Figure 10 shows 8-side circle example.
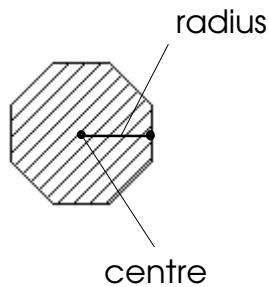
radius

centre

Figure 10 Example of a Circle

- **ring.** It's the object, which is specified by its centre, radius and width. The ring is represented in LW as wire, because the ring width and end margin will be specified by the wire width and the end margin in the moment of creation. The ring width may be equal to zero - then the ring will be represented as line in the editor database. Figure 11 shows the ring example.
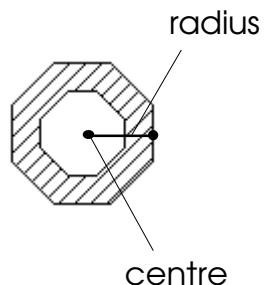
radius

centre

Figure 11 Example of a Ring

- **arc.** It's the object, which is specified by its centre, radius, angle between the extreme points and width. The arc is represented as wire in LW, because the arc width and its end margin will be specified by the wire width and end margin in the moment of creation. The arc width may be equal to zero - then the arc will be represented as line in the editor database. Figure 12 shows the arc example.

radius

end angle   start angle

centre
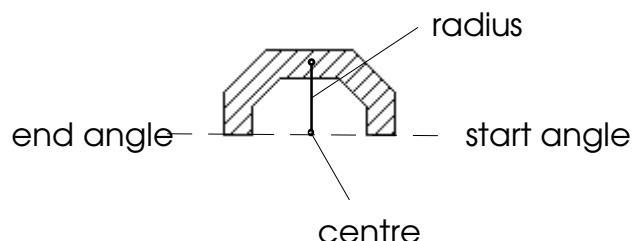
Figure 12 Example of an Arc

- **text.** Objects of this type are specified by the layout font, its height, orientation and base point. By default the text height is equal 100. The layout font is proportional. When you change the height of text, its width is also changed. You may create its own layout font - there is explained in *Section 7. Miscellanea. The creation of layout fonts*. Figure 13 shows the text example.

base point

Figure 13 Example of a Text String

For the arc primitives (circle, ring and arc) there is one more parameter - the number of the sides of the approximated line, because these primitives are represented in LW database as polygons and wires. This parameter also is named the number of arc segments

Layers

All primitives must be placed on any of the layers. In LW all layers are the same, except one, in which only instances can be placed. This layer is named CALL in CIF format and 00 in SOURCE format.

No more than 48 layers can be used in the standard version of LW (including the layer CALL-00), but you can order a version of editor with desired number of layers.

Each layer can be both visible or invisible. Visible layers also can be in active state. Each layer can have its own color, fill pattern, line pattern, which can let you to distinguish it from the others.

All layers are collected into the layer table. Layer table is loaded to LW from the special file, which you must create yourself. The full information about every layer is written in this table. The example of layer table is contained in *Appendix B. Sample layer table file*.

Properties

All the basic primitives without exception can have a property. The property is a text string with arbitrary

contents. In LW properties can be used in the checking the electrical circuit and in design rules checking. When you set a property, you must comply with standards, described in *Section 4. Advanced Course. Design rules checking.*

Cells and instances

Cell is the basic structure in design. You can design a cell as complex piece of layout. More complex designs are usually broken into smaller cells and referred from the main cell as instance.

Each cell must have its unique name, which is also its file name, in which the cell is stored. The cell name must be different from all the other cell names to distinguish data.

Before working with the cell you must open it. Opening cell means loading cell data into the LW database. After completing cell image will be shown in the current window.

You may edit cells have been loaded into database. Unlike the other editors there are no single active window and single active cell. In other words, all windows are active. After you have chosen the operation the editor will demand to choose the window, in which this operation will be done.

When you have finished session you may store the cell in file. Now you can store data in two formats: CIF and SOURCE.

Hierarchy

Design hierarchy in LW is accomplished by instantiating one cell within another. Instantiated cells may themselves contain instances of other cells. When you instantiate a cell you create a reference to that cell in the layout. Editor marks and labels the area the cell takes up in the layout, but does not "copy" the cell contents into the layout.

You can place the instance of cell, which already is loaded into editor database. Alternatively, you can place the instance of cell, which is not loaded into database. This flexibility lets you to realize both top-down design and bottom-up design.

When you place the instance of cell, which does not exist in database, the cell is not automatically loaded from file. When the editor can't find cell in database, it creates new cell.

Editor's database

Working with LW, all project data are usually stored in one directory on disk. Project data can be stored in two formats: CIF and SOURCE. Different variants of format determine different set of project files. When the project data are stored in CIF and SOURCE formats every cell of project is stored in its own file. When the project is stored in CIF 2.0 format, all data are saved to one file. Figure 14 shows difference in the storing data files depending on format.

project in database   **project files**   **project file**
                          in CIF or SOURCE format   **in CIF 2.0 format**
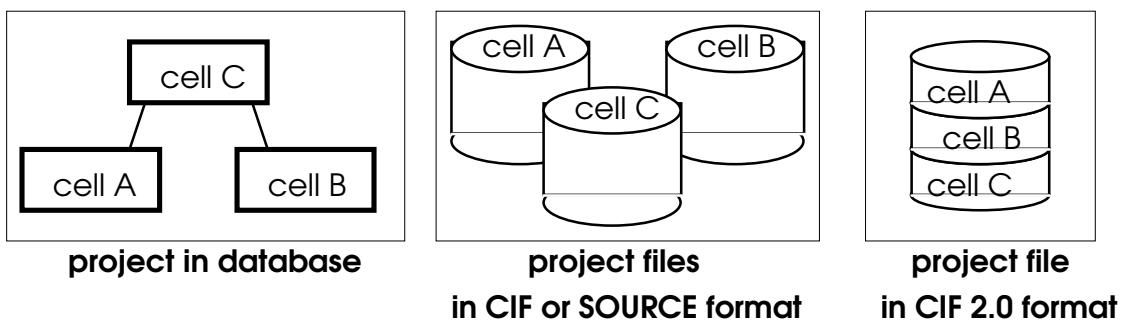
Figure 14 Difference in the Storing Data Formats

Userware

The term userware refers to the interface between LW and you, the user. This userware includes the set of windows, control devices (mouse and keyboard), fonts and configure files.

Mouse

In LW mouse is used for:

- entering co-ordinates

- selecting the windows

- picking menu items and control command

All operations, which are described above, are started by the <u>mouse click</u>. Clicking a mouse button means to press and quickly release the button that makes a click noise. When you click the mouse button you must not move the mouse.

The mouse clicks by different buttons have different sense:

- **left** button selects objects, windows, menu items, turns on/off menu switches and enters the co-ordinates.

- **right** button repeats the last menu command or the last window control command.

Keyboard

In LW keyboard is used:

- for providing the dialogue with editor

- for windows control

- for entering co-ordinates

- for selecting windows

- for picking menu items and control command

Moreover, if the editor is installed on IBM PC, you may manage the cursor through the keyboard.

Pressing the ENTER key or SPACE key have the same effect as the clicking the left mouse button. Because, if you have not a mouse in your computer, you can use the ENTER key or the SPACE key.

The dialog window at the bottom of screen is used for dialogue with the editor.

To control cursor through the keyboard you must turn on the SCROLL LOCK mode at first. When you turn on this mode the SCROLL LOCK indicator (to the right on the keyboard) must be highlighted.

To manage the cursor you can use cursor control keys on the keyboard. When you press one of these keys once, the cursor will be moved in the direction mentioned on the pressed key. You can set the step, on which the cursor is moved when single pressing the control key. By default the step is equal to 8 pixels.

To change the cursor step use **"+"** and **"-"** on numerical part of the keyboard. Single pressing of **"+"** (**"-"**) key increases (decreases) the cursor step by 8 pixels. The value of minimum step is equal to 8 pixels also.

To move cursor on one pixel, press SHIFT and control key you need simultaneously.

If you do not know the sense of some menu command or window control command, move the cursor to this command and press the F1 key. After that LW displays the help information in the dialog window.

Windows

In LW there are some types of windows. The principles of its managing are similar.

Layout window

A layout window is used for entering and editing the project layout. There can be several layout windows in LW. Figure 15 shows the sample layout window.
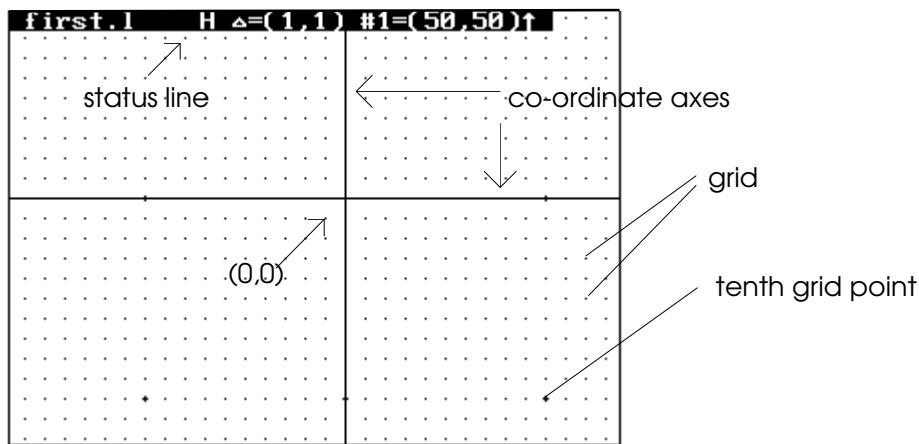
Figure 15 Sample Layout Window

At the top of window there is a status line, which contains information about this window and about cell opened in this window. There are co-ordinate axes in the layout window, which are highlighted usually by white color. The intersection point of axes has co-ordinates equal to (0,0)

You can set up to 9 different auxiliary grids with the various steps for each layout window. At the same time only one of the grids can be shown in window. Grid can be shown over the layout and under the layout as well.

Besides the auxiliary grids there is a round grid, which is used for co-ordinate rounding. All primitive's co-ordinates are snapped to grid (i.e. all entering points will be aligned to the grid junction).

You can zoom image and flip it on horizontal and/or vertical directions in any layout window.

Two co-ordinate systems are connected with the layout window:

- **pixels** co-ordinate system of screen. The boundaries of all windows are connected with this system.

- **applied** co-ordinate system of window. The boundaries of shown cell piece, image scale and view modes are connected with this system.

The status line starts with the cell name, which is shown in current window. Further, symbols and its combinations follow the cell name:

- **the small star "★"** in first position after the cell name indicates, that the cell is modified and it's necessary to update it. When you update this cell, small star disappears.

- **symbols X è Y** in second and third positions show the mirror view modes on axes X è Y correspondingly (look from behind it).

- **the number** in fourth position shows the shown hierarchy depth. If the symbol **"H"** is placed instead of number, it means, that full cell hierarchy is shown in the window. In the other cases only current cell is shown in window.

- **the round grid step** on X and Y axes is placed after the hierarchy depth. This is a following string:

    Δ=(*step_on_X*,*step_on_Y*).

    Values of the grid step on both axes are in parenthesis.

- **the information about auxiliary grid**: expression of the following type:

    #*grid_index*=(*step_on_X*,*step_on_Y*)

    where *grid_index* is the index of grid, which is shown currently. Values of the grid step on X and Y axes are in parenthesis.

- **up "⇑" or down "⇓" arrow** at the end of line show, how the grid is drawn relatively to the image (over the image or under the image). If the arrow is set upwards (**"⇑"**), then the grid is drawn over the image.

You can turn off the auxiliary grid and co-ordinate axes from the layout window. In this case the information about auxiliary grid and it draw mode will be absent in the status line.

Dialog window

Usually the dialog window is placed at the bottom of screen and provides the dialogue between user and editor.

When you need to enter some string through the keyboard,
the editor displays corresponding instruction and current
value (highlighted) of this expression in the dialog
window. To change the value of expression, use "⇦" and "⇨"
keys for cursor moving and DEL and BACKSPACE keys for
deleting the characters.

To change the character map on alternative (national) map,
presses both ALT keys simultaneously. To return backward
use also the ALT keys. To enter single alternative
character (i.e. for example, if you are in the English
keyboard mode and want to enter the national symbol) press
the ALT key and the desired key simultaneously.

After completing the string press the ENTER key or click
the left mouse button to enter text.

Menus

Usually the menu window is placed in the left and upper
corner of screen. In LW there are several variants of
menus, one of these is the main. The menu title, which
shows the current menu variant, placed at the top of menu.
Menus contain four types of items:

- items, which change the menu variants (menu
  switchers)

- switches

- commands of continuos action

- auxiliary commands of single action
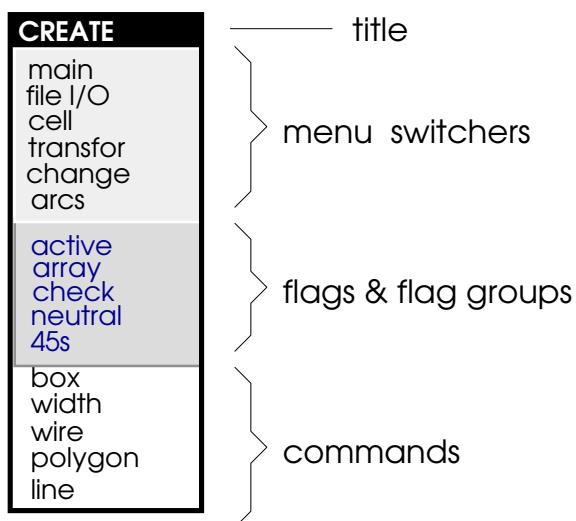
figure 16. Menu Example

To pick the menu item, click the left mouse button or press
the ENTER (SPACE) key on the item.

Menu switchers

These items are placed in the upper part of menu and are highlighted by the black color on the grey background. After you pick one of these items the current menu variant is changed and a new menu variant will be drawn in menu window. The menu title is changed too. To return backward pick the **"back"** menu item (it is highlighted by the red color). In some menu variants the **"quit"** item is substituted on the **"main"** item. After you pick the **"main"** item, the current menu variant will change to the main.

Switches

Switches are placed in the middle part of menu. They are highlighted by the blue color on cyan background. Switches are the following:

- flags

- flag groups

After you pick the flag switch, this flag is set active and menu item is highlighted by the bright green color. When the picking is repeated, this flag becomes non-active and the item color is set cyan again.

After you pick the item, which is the flag group, the value of this item is changed to the next in flag group.

Warning: almost every flag group contains a flag(s), which assumes you should be careful while setting it. When these flags are selected, they are highlighted by the bright green color.

Commands of continuos action

These commands are placed at the lower part of menu and highlighted by the white color on the blue background. Usually these commands are layout commands. After you pick one of these commands, this command becomes active and is highlighted by the bright green color. At the same time there is only one active command.

When active command is done, it displays the help instructions in dialog window.

To reset the active command click the left mouse button or press the ENTER key or the SPACE key on the field of command.

To switch active command move the cursor to command you want to set active and pick it by clicking the left mouse button or by pressing the ENTER key or the SPACE key. After

that, previous active command becomes non-active and its color becomes white again.

When the current menu variant is changed, the states of commands and switches are stored and, when you return backward, those states are restored.

Commands of single action

These commands are similar to the commands of multiple action. The are also placed in the lower part of menu. The difference between these types of commands is that the commands of single action are executed once and just while picking.

The window of current co-ordinates

Usually the window of current co-ordinates is placed in the left lower corner of screen. It is used for displaying the last entered co-ordinates (X, Y). Except the current co-ordinates in this window you can see offset that is relative to previous entering co-ordinates and distance between these points.

Warning: All values are in the applied co-ordinate system of corresponding layout window.

The layer window

The layer window is placed to the left on the screen and is used for displaying and managing layers. The information about layers is in <u>layer table</u>, which can be shown in the layer window both in the vector form and in the matrix form (if the window size allows it). The layer table contains the following data about every layer:

- color and fill pattern of the layer

- line style of the layer

- layer visibility at the current moment

- layer activity at the current moment

- layer identifier in CIF and SOURCE formats

In layout windows every layer is placed on its level. Primitives placed on the upper level layers will overlap those ones placed on lower level layers. The order, in which the layers are distributed on the levels, is reflected in placing the layers in the table. The most higher layer is placed in the first row and the first column of the table. The next layer is placed to the right of the first (or below it if the table is a vector), etc.

When you make the mirror reflecting the image in the layout window, the upper layers will be found underneath and the lower layers will be found overhead.
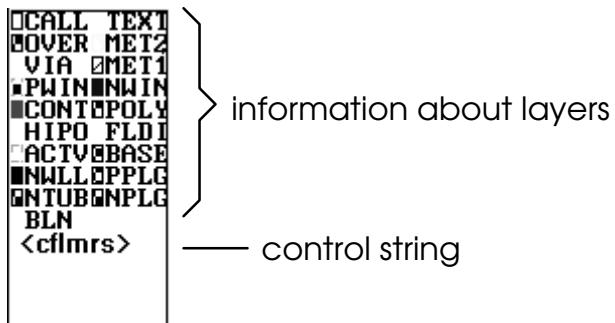
Figure 17 The Layer Window

There is the control string at the bottom of window. It allows you to control the layer table. To pick the command from this string, move the cursor to the symbol assigned with this command and click the left mouse button. After that you must select the layer, which you want to manage, and command will be executed at once.

There is another way to enter the command: you can press the keyboard key, which is corresponded to the command, on the selected layer - the command executed with the same effect as in first way.

In the layer window you can change the layer color, fill pattern and line style of layer, it's placing in layer table. You can also store and read the layer table from file with the help of command string.

For more information about the layer window see the *Section 4. Advanced Course. Layer table control*.

Control window

The control window is used for managing all the editor's windows and image in layout windows. There are several symbols in control window and each of them is assigned to some control command. You can enter the control command by two ways:

- by clicking left button of the mouse (or pressing ENTER key or SPACE key) on the necessary symbol in control window. After that the editor offers you to select the window, in which the command will be done.

- by pressing the keyboard key, which is corresponded to the symbol, on the window, in which you want the command to be executed. Functional keys F1..F12 are corresponded to some control commands too.

```
?▦XYH▲◀↑▶▼−+WG#
^CVMSRF
```

Figure 18 The Control Window

Warning: When entering the command through the keyboard, remember, that the symbols, corresponding to control commands, are uppercase letters. Do not forget to press CAPS LOCK or SHIFT when it needed.

When using the control commands, you can change the window size, move it, change window colors, store the window configuration in file, etc. You can also perform some operations with the image in the layout windows: change the image scale, scroll image both on horizontal and on vertical, place and dispose the auxiliary grid, etc. For more information see the *Section 4. Advanced Course. Managing windows*.

Flexible editor's interface

The editor is composed in such way, that you can customize its interface. You can:

- change the window configuration

- change the fonts used in LW

- change fill patterns and line styles

- change the color palette

- change the editor text reports and menu item names

For that you must change information in the configure files. These files are in ASCII format and you can easily change them. The main configure file is lw.win. For more information see *Section 7. Miscellanea*. *Managing LW interface*.

Edit concepts

LW has several commands for objects the editing. You can copy the objects, remove them, change their shape, multiply them (create an array based on the selected primitive), transform the objects (rotate them, flip).

To modify objects, you have to <u>select</u> them first and then enter a command to change them. The contour of selected objects will be highlighted (usually by white color). The order of object selecting is regarded in the next chapter.

Selection

Before object selection you must set list of layers, on which you can select objects. There are three variants of layer list:

- there are only active layers in the list

- there are only visible layers in the list

- all layers are contained in the list

There are three ways of object selection:

- **selection by point.** In this way you select the objects by clicking the left mouse button or pressing the SPACE key or the ENTER key. All primitives surrounding the current cursor position will be selected.

- **selection by area.** In this way you must move the cursor to one corner of the area you want to define and enter current point (by clicking the left mouse button or pressing the ENTER key or the SPACE key).

  After that when moving the cursor, LW creates "rubber" rectangle. One of the corners of this rectangle is pointed in the last entered point. The opposite corner moves with the help of cursor.

  When you repeat the entering the current point, the "rubber" rectangle will disappear and all objects, intersected with this rectangle or surrounded by it, will be selected.

- **global selection.** In this case you select all objects, which can be selected.

Section 3
Sample Edit Session

Starting edit session

Before starting LW it is necessary to be convinced the layer table file with extension *.ltb (by default lw.ltb) exists. This file contains the whole information about layers, which can be used in the edit session. If you start LW, when this file does not exist, abort the edit session by pressing CTRL-C or you will have some troubles.

To start LW type in the command line:

> **lw**

and press the ENTER key. Just picture, shown in Figure 19, appears, you can begin to work with editor.
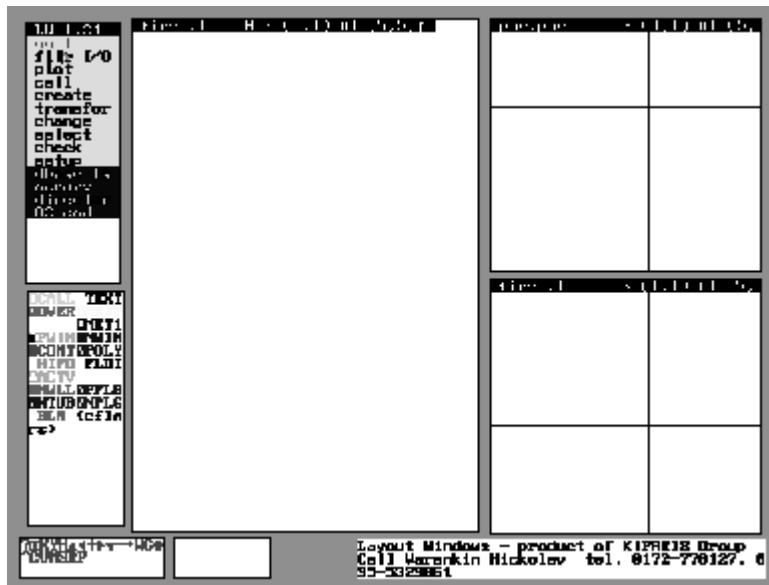
Figure 19 The Rough Editor's View

After ending the edit session, the window configuration is automatically stored in the file named lw.win. For layout windows their grids, cell name, view bounds and view modes are stored too. Thus, all information about cell view belongs to the window. When LW is started again, the window configuration and all data about cells and their view are restored, but the cells are not loaded into the database automatically. That is why when LW starts, all layout windows will be without image.

Opening cell

To begin editing, you must open the cell. When the cell is opened, its layout information is got from editor's database and displayed in the selected window. The cell can be opened only in existing window. If the cell is not exist in the database, the editor searches the file with the cell in early specified directories and then loads it into the database. If the cell file is not found, then the cell is opened as new cell.

If the loaded cell contains an instance, then all cell hierarchy beginning with the current cell is loaded into the database. You can limit the depth of hierarchy. In detail it discussed in the *Section 4. Advanced Course. Opening the cell.*

If a new cell is opened, then the normal view modes of image will approximately set and the image bounds will be distant from the point (0,0) on 50 units.

When the existing cell is opened, the bounds of viewing cell piece, the image scale and view modes in the window are remained former (before opening).

You can set the mode, when the whole opened cell will be displayed in window. In detail it is discussed in *Section 4. Advanced Course. Opening the cell*.

To open the cell, switch to FILE menu and activate the **"read"** command (by default it is already activated). Then you must select window, in which the cell will be displayed. The editor prompts you to enter the cell name:

Read file *file_name*

where *file_name* is the name of cell, connected with selected window. If you want to open another cell, enter the cell name instead of *file_name*. Finish the entering by pressing the ENTER key. After that the cell will be loaded into the database and displayed in the selected window.

Drawing layout

Before drawing you first need to define the layer (or layers) you want to place primitives on.

Commands to drawing basic primitives are contained in the CREATE menu, text and instances are created with the commands from CELL menu, other primitives are drawn with the ARCS menu commands.

After you switch to necessary menu variant, draw command and necessary switches are set to the previous state. Then

you must follow the editor's instructions - and the
primitive will be drawn.

After completing drawing object active command and states
of switches are not changed. The editor is ready again for
drawing the primitive of current type and displays
corresponding instructions. To draw the primitive of
another type, change the active command.

Editing layout

To modify objects you create, you must select them first
and perform a command to change them. Edit commands are
contained in the CHANGE, TRANSFORMATION and SPECIAL
TRANSFORMATION menu variants.

You can select objects by two ways:

- with commands from SELECT menu. After you have
  selected objects, you must return to previous menu
  variant, set the **"point"-"area"-"selected"-"global"**
  switch to the **"selected"** state and select the window,
  in which you want to execute the edit command.

- in the current menu variant. After you pick the edit
  command, set the **"point"-"area"-"selected"-"global"**
  switch to the state you want (except **"selected"**) and
  realize the selection. Continue the edit command with
  selected objects.

Storing data

After the cell editing is finished you can store it in
file. To store the cell, switch to the FILE menu, set the
"write" command to the active state and select window, in
which this cell is displayed. After that the editor prompts
you to enter the storing cell name:

Write file *file_name*

where *file_name* is the cell name displayed in selected
window. This name also is the name of file, in which the
cell will be stored. If you want to store the cell with
another name, enter the cell name you want instead of
file_name. Finish the entering by pressing the ENTER key.
After that the cell will be stored in file.

Warning: when you store the cell in the file with another
name, the cell name in the database is not changed.

Ending edit session

To quit from LW, pick the **"quit"** command in the MAIN menu.
If at teast one cell is not updated, the editor displays
the following string in dialog window:

    Really quit?(y/[n])

To cancel the quit, enter the **"n"** character (or press
ENTER at once). In opposite case enter the **"y"** character
and press the ENTER key. After that the edit session is
finished.

Section 4
Advanced Course

Creating layer table file

Before starting the editor, you must create the layer table file. This is ASCII file and it has the *.ltb extension. The name of layer table file is saved in editor configure file (usually the layer table file name is lw.ltb). When the editor starts, the information from layer table file is loaded into editor's database and used in further. In learning LW it mostly to use the file named lw.ltb. The LW installation procedure creates file with same name. The following listing is the sample layer table file:

```
4 layers, 4 in use:
(mode=2,color=10,filling=1,style=0)=00=CALL
(mode=2,color=12,filling=7,style=0)=01=LAY1
(mode=2,color=4,filling=7,style=0)=02=LAY2
(mode=2,color=1,filling=7,style=0)=03=LAY3
```

The format of layer table file is following:

1. The first line contains information about maximum and used layer count:

```
4 layers, 4 in use:
```

The maximum layer count must be equal to used layer count.

2. The information about each layer is contained in the parenthesis:

```
(mode=2,color=4,filling=7,style=0)=02=LAY2
```

where

- **mode** is the layer mode. When the layer table is stored, the editor saves the current state about layer to this field. When you create the layer table file, use:

```
mode = 2
```

- **color** is the index of layer color. See color indexes and their RGB values in the *Appendix A. Fill patterns, line styles and colors.*

- **filling** is the index of layer fill pattern. See fill patterns in the *Appendix A. Fill patterns, line styles and colors.*

- **style** is the index of layer line style. See line styles in the *Appendix A. Fill patterns, line styles and colors.*

3.   Each layer description is finished with layer names:

- **02** is the layer identifier in the SOURCE format.

- **LAY2** is the layer identifier in the CIF format.

In any variants the layer table must contain the layer with identifiers CALL (in the CIF format) and 00 (in the SOURCE format). Its color, filling and style may be arbitrary.

Starting LW

You can start LW with several switches:

> **lw** [**/w**[*configure_file*]] [*geometry*]
   [**/CIF** | **/CIF2.0** | **/SOURCE**] [*file*] [*file*]...

where

- **/w** is a switch, which allows you to set the window configuration, which contained in the file named *configure_file*.

  If **/w** switch is set without the file name, you must place windows in the screen without assistance. When the editor starts, only menu and dialog window will appear in the screen. In dialog window the editor will display instructions for placing other windows. To do this:

  1. Move the cursor to one corner of the window you want to define.

  2. Enter the position of corner by clicking the left mouse button (or pressing the ENTER key or the SPACE key).

  After that when the cursor is moved, the "rubber" rectangle will appear in the screen. One of the corners of this rectangle is positioned in the point you enter. The opposite corner is moved with the cursor.

  3. Move the cursor to opposite corner of the window you want to define.

4. Enter the position of corner by clicking the left mouse button (or pressing the ENTER key or the SPACE key).

After that the window will be drawn in the screen.

New configuration is stored in file named lw.win and it can be easily changed in further.

- *geometry* is the text string, which specify the size and placement of the area in screen, in which editor's windows can be formed. The geometry is specified relative to screen bounds. If you not specify the geometry, all screen area will be used. The geometry is one of the text strings (without spaces). All formats are similar one used in X Window System. First format is supported for compatibility:

  -g *area_width* õ *area_height* ± *hor_offset* ± *ver_offset*

  or

  = *area_width* õ *area_height* ± *hor_offset* ± *ver_offset*

  where

  - *area_width* and *area_height* specify the area size (in the pixel units).

  - *hor_offset* specifies the horizontal offset of area relative to left screen bound (if there is the **"+"** character before offset) or relative to the right screen bound (if there is the **"-"** character before offset).

  - *ver_offset* specifies the vertical offset of area relative to upper screen bound (if there is the **"+"** character before offset) or relative to the lower screen bound (if there is the **"-"** character before offset).

  The following example shows sample geometry for full VGA (IBM PC) screen:

  -g 640õ480+0+0or   =640õ480+0+0

  If specified geometry does not allow the editor to create windows, which are specified in the configure file, the editor switches to the interactive mode to create other windows.

- **/CIF**, **/CIF2.0**, **/SOURCE** are switches, which specify the file format when automatic file loading is done. By default the **/CIF** switch is assumed.

- *file* is the file name (with extensions, if it exists) for automatic loading. There can be several file names with spaces in command line. The file name can contain the path to the file. In file name you can't use wildcards**"*"** and **"?"**.

  The formats of all automatically loaded files must comply with the specified format switch.

All arguments and switches may be specified in arbitrary order.

Opening the cell

To open a cell:

1. Switch to FILE menu

2. Set the **"read"** menu command to the active state

3. Point the window you want to load and view the cell

Before the **"read"** command is activated, you can set modes for cell reading; it discussed in detail below.

If you select window, which already contains a cell, the editor displays the following string in dialog window:

   Delete current file in database? (y/[n])

If you answer **"n"**, the cell will remain in the database. Else, if you answer **"y"**, the cell will be removed from the database. Only current cell will be removed; cells instantiated to the current cell will remain in the database.

Further, the editor displays the following string:

   Read file *file_name*

where *file_name* is the name of cell, which the editor gets from window status line. If you press the ENTER key, the editor will load the file with the cell named *file_name*. Thus, if you want to load another cell, you must first type new cell name and then press the ENTER key.

If the cell is already exist in the LW database, the editor asks:

   Load from database? ([y]/n)

If you answer **"y"** (or press the ENTER key at once) the cell from the database will be shown in the window. Else, if you answer **"n"**, the information about cell will be removed from the database and the cell will be reloaded from the file.

After listed questions the editor starts reading the cell and all cells instantiated to it. Each time cells are read from disk, if they are not exist in the database.

If the file with opened cell does not exist, the cell is opened as new and the editor will report in the dialog window:

New file

In other cases reports about read file and its hierarchy level will be displayed in dialog window. Reading every 200 (500 for OS UNIX) primitives is indicated by the special symbol (small face **"☺"**) in the dialog window.

If some number of read error is founded, the editor asks you:

Too much errors. Continue? (y/[n])

To cancel file reading, answer **"n"** and the already loaded information about the cell is removed from the database. Else, when you answer **"y"**, the editor will continue to read this file until the some number of errors will be found or until the end of file will be found.

After the end of file is reached, the editor displays the following report:

File is successfully read

and after that you can work with loaded cell.

Loading hierarchy of cell

You can limit the hierarchy depth of loaded cells. For that, before executing the **"read"** command set the **"levels"** flag to the active state. Then the editor displays the following string in dialog window:

Hierarchy read depth *hierarchy_depth*

where *hierarchy_depth* specifies the hierarchy depth of loaded cells. By default the *hierarchy_depth* is equal to zero, i.e. there is only current cell will be loaded into database. To change it, enter the necessary value instead of zero and press the ENTER key.

When the **"levels"** flag is passive, the all cell hierarchy is loaded.

Loading files in CIF and SOURCE formats

Current editor version can read and save cells to files only in two formats: CIF and SOURCE. To set necessary format, set the switch **"CIF"-"SOURCE"-"CIF2.0"** in the FILE menu to the value you want (to set value, pick this menu item until necessary value will be shown). Before reading existing cell set the format switch to the appropriate value.

Load switches

1. When the file is being loaded and error limit exceeded, the editor breaks reading and displays report in dialog window:

    To much errors Continue? (y/[n])

    You can specify the limit of read errors. For that, set the **"err.lim."** flag to the active state. After that LW prompts you to set the limit of read errors:

    Error limit *error_limit*

    where *error_limit* is the current limit of read errors. By default the *error_limit* is equal to 1. Type value of error limit and press the ENTER key (or press the ENTER key without value entering, if the current value satisfies you).

    If the **"err.lim."** flag is passive (equal to zero), the editor will not break until it reaches the end of file.

2. If the cell is just loaded, the view bounds of image remain by previous and only piece of cell may be shown. To show all layout of cell set the **"win.frame"** flag to the passive state before executing the **"read"** command.

3. If the file with cell to open is not in the current directory, in order to find it you must specify the path (list of directories) to that directory. To do this, set the **"path"** flag to active state. After that the editor prompts you to specify the directory list:

    Enter directory list to search files

    Type the directory list, which contains cell files you need. The list format must be same as your operating system use. To separate directories, use spaces, colons or semicolons.

Storing the cell

To store an existing cell in file:

1. Switch to FILE menu.

2. Set the **"write"** command to the active state.

3. Select the window with cell, which you want to store

Before the **"write"** command is activated, you can set modes for cell storing; in detail it discussed below.

LW prompts you to enter the name of stored cell:

Write file *file_name*

where *file_name* is the cell name, which the editor gets from the window status line. If this name satisfies you, press the ENTER key. In opposite case enter cell file name you need (the file name in database will not be changed).

If the file is already exist, the editor asks:

overwrite? - (y/[n])

If you answer **"n"** (it is set by default) storing will not be executed. If you answer **"y"**, the file will be overwritten.

After above questions the cell will be stored in the file. At the end of storing, if it is successfully, the editor displays the following report:

File successfully saved

In opposite case the report about write error(s) will appear in dialog window.

Storing hierarchy of cells

In LW there are three variants of storing cell hierarchy, to which the various states of **"current"-"flatten"-"hierarchy"** are corresponded:

- In the **"current"** state only current cell is stored

- In the **"flatten"** state all subcells are virtually flattened to the current cell and whole information is saved into one file.

- In the **"hierarchy"** state the all cell hierarchy is stored beginning the current cell. Each cell is stored in its own file.

Saving files in CIF and SOURCE formats

To save the cell file in the file format you want, before activating the **"write"** command set the **"CIF"-"SOURCE"-"CIF2.0"** flag switcher to the corresponding state.

When the cell is stored, the information about the file format is not saved to the file. Thus you must remember, in what format files of your project were saved.

Warning: when the file format and format, which is set in the editor, are not equal, when reading data the editor will display reports about read errors, but it will not display report about different formats.

Store switches

1. The **"all"-"current"-"visible"** switcher specifies the layers, which can be stored in the file:

   - In the **"all"** state all existing layers will be stored.

   - In the **"active"** state all layers, which are active at the moment of storing, will be stored.

   - In the **"visible"** state only visible at the storing moment layers will be stored.

2. The **"global"-"point"-"area"-"selected"** switcher specifies the select mode of primitives, which will be stored. Only primitives, placed on the layers corresponding the state of the **"all"-"current"-"visible"** switcher, will be stored.

   - In the **"global"** state all objects, which can be stored, will be stored.

   - In the **"point"** state objects, which can be stored, are selected by clicking the left mouse button or pressing the ENTER (SPACE) key.

   - In the **"area"** state the stored objects are selected by the "rubber" rectangle. The corners of this rectangle are set by clicking the left mouse button or pressing the ENTER (SPACE) key.

   - When objects are selected by command from SELECT menu, the switcher must be set to the **"selected"** state.

3. The **"scale"** flag specifies the zooming coefficient of primitive's co-ordinates in the storing cell. When you activate this flag, the editor offers you to enter zooming coefficient. This coefficient is entered as proper fraction: first the numerator of fraction, the denominator must follows it. The space between numbers also must be placed. For example, if the zooming coefficient is 0.01, enter the next string:

   > 1 100.

4. When the **"box>pol"** is in the active state, all boxes in the cell are stored as polygons.

5. When the **"wire>pol"** is in the active state, all wires in the cell are stored as polygons.

6. If the **"confirm"** is in the passive mode, when the **"read"** command executes, the editor stores the cell without asking you the cell name and overwrite permitting of the cell

Ending the edit session

To quit from LW, pick the **"quit"** command in the MAIN menu. If some cells are changed and not updated, the editor lists unsaved cells and asks you:

> Really quit?(y/[n])

To cancel the quit, enter the **"n"** character (or press ENTER at once). In opposite case enter the **"y"** character and press the ENTER key. After that the edit session is finished.

Managing windows

You can manage the LW windows by the commands from control window. Besides that, commands of layout image control also are in control window.

To create and/or deltete layout windows, use the commands from SETUP menu.

There are two ways for entering the commands from control window:

1. **a)** Select the command from control window.

   b) Move the cursor to the window, by which you want to control, and select it.

   c) Further, follow the editor's instructions.

2.   **a)**   Move the cursor to the window you want to control.

b) Press the key or some keys, which are correspond to the command you need.

c) Further, follow the editor's instructions.

Functional keys F1..F12 are corresponded to some commands from control window. When these commands will be discussed, functional keys will be placed behind symbols correspoinding control commands.

For some operations in the layout windows (image scrolling, switching the grid) **"1".."9"** keys may be used for effect reinforcement or for other actions. These keys are used by following way:

1. Move the cursor to the symbol corresponding to the command you need.

2. Press one of **"1".."9"** keys.

The symbol corresponding to the last executed control command is shown inverted in control window. For repeating the last command click the right mouse button on the window you want to control it.

Common operations

Moving a window

The **"M"** symbol is corresponded to the window moving command.

To move window:

1. Select the moving command and the window, which will be moved.

2. Select the base point of window.

After that when cursor is moved, the "rubber" line will follow it. This line is started from the selected base point and shows the moving direction and distance between old window position and new window position.

3. Select the new position of window base point in the screen.

After that, the "rubber" line disappears and the window is moved to the new position.

Changing window size

The **"V"** symbol is corresponded to the command of window size changing.

To change window size:

1. Select the command "V", next window to be changed.

2. Set the new window size by the "rubber" rectangle.
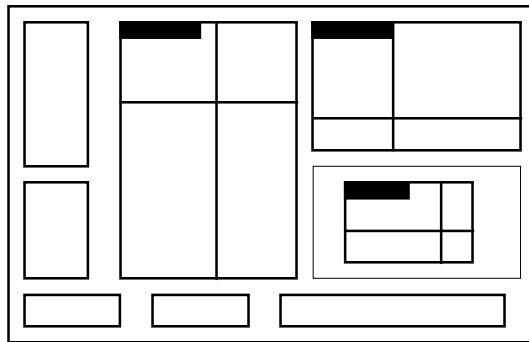
Figure 20 Set the New Window Size by the "Rubber" Rectangle

After that the window size will be changed and it will be redrawn with new size.
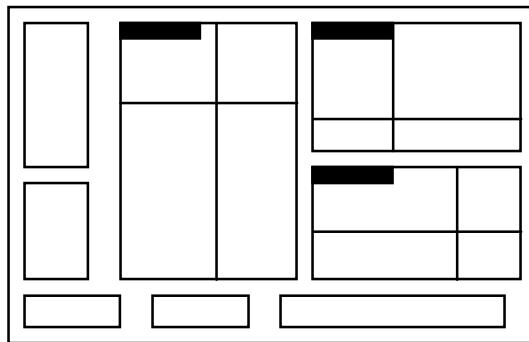
Figure 21 The Window Size Will Be Changed and It Will Be Redrawn With New Size

The window size not changes in two cases:

* if the window overlaps other windows

* if the window size is too small

In both cases repeat the setting the window size.

Window redrawing

To redraw windows, you can use two symbols:

- **"R"** (F11) is used for redrawing one (selected) window.

- **"F"** is used for redrawing all windows. In this case window selecting is not required.

Window color setting

Three colors corresponds to any window:

- background color

- foreground color

- highlight (frame) color

You can change colors of any window by using the **"C"** symbol.

After you enter the command and select the window, the current indexes of window colors will be displayed in dialog window. Color indexes will be displayed in following order: first the background color index, after this - foreground color index and at the end - the highlight color index. You can change them using keyboard. When the entering is finished, press the ENTER key. The window will be redrawn with new colors.

Storing window configuration

To store the window configuration, use the **"S"** (F12) symbol. The window configuration is stored in the current configure file (in the lw.win or in the file, specified by /w *file_name* switch in the command line when LW was started).

Creating an additional layout window

In LW you can create several windows, in order to show or edit the same cell(s). To create an additional layout window:

1. Switch to SETUP menu and select the **"subwindow"** command.

2. Select window with the source cell, which will be shown in the new window.
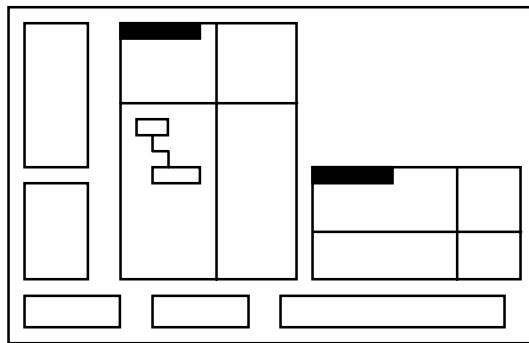
Figure 22 Select Window With the Source Cell, Which Will Be Shown in the New Window

3. Set the new window sizes by the "rubber" rectangle.

After that, the additional window will appear in the screen and the selected cell will be shown in this window. Image view bounds will be same as in the window, where you select the cell.
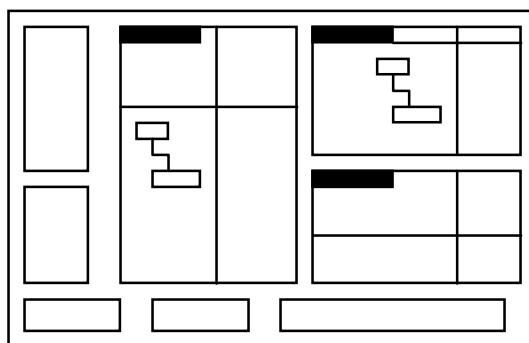
Figure 23 The Additional Window Will Appear in the Screen and the Selected Cell Will Be Shown in this Window

Creating a new layout window

To open a new layout window:

1. Switch to SETUP menu and select the **"window"** command.

2. Set the created window sizes by the "rubber" rectangle.

After that the new layout window will appear in the screen. By default the cell named unknown.l will be opened in this window.

Closing a layout window

To close a layout window:

1. Switch to SETUP menu and select the **"delwindow"** command.

2. Select the closed window.

After that the selected window will disappear from the screen

Note, closing window does not mean deleting cell from database.

Managing grid in layout windows

You can connect with any layout window up to 10 grids: one (zero) is the round grid and other nine grids are visible grids. At the same time only one visible grid can be shown in the layout window. The round grid is never shown in the layout window.

There are several commands for grid managing:

- **"#"** (F6) allows you to specify the current grid step. After you enter this command and select the window, the editor displays current grid index and it steps on X and Y axes in dialog window. If it satisfies you, press the ENTER key, else first enter new values.

  For setting another auxiliary grid use **"1".."9"** keys, when you select command in control window.

  The current grid is not shown in the window, if the distance between grid nodes is less than 5 pixels.

- **"^"** (F7) allows you to specify the round grid step. The sequence of action is the same as when the **"#"** command is done.

- **"G"** (F5) switches view modes of grid and co-ordinate axis relative to the layout:

  - ★ without grid and co-ordinate axes

  - ★ grid over the layout

  - ★ grid under the layout and view

  In last two cases the co-ordinate axis will be shown in the layout window.

Managing layout image

All commands of image control (exactly, symbols corresponding them) are placed in control window. The following command list contains all command of image control, its description, symbols and functional keys for activating them.

- **"┼"** (F8) allows you to set the image centre in the window. After you select this command and the window, move the cursor with rubber line to the centre position you want and enter current point (by clicking the left mouse button or pressing ENTER or SPACE keys).

- **"⇦"**, **"⇨"**, **"⇧"**, **"⇩"** allows you to scroll image on four directions. When you once enter one of this command, the image is scrolled on the 3/4 of view size.

  You can make this commands more powerful using the **"1".."9"** keys. Distance, on which the view will be moved, will be equal to the view size multiplied by the number on the pressed key.

  For example, if you press the **"8"** key on the scroll symbol, the layout image will be scrolled on the 8*3/4=6 of view size.

  This coefficient is stored, and in further the image will be scrolled on the same distance.

- **"+"** (F10), **"-"** (F9) allows you to change the view scale of image (**"+"** increase the value of scale and **"-"** decrease the value of scale). When zooming cursor position points to new image centre. The image scale will be changed 1.5 times.

  You can make this commands more powerful using the **"1".."9"** keys. The scale to be set will differ from the current one by the amount equal to pressed key.

  This coefficient is stored, and in further the image will be zoomed (magnified or diminished) as same.

- **"W"** (F4) allows you to set the view bounds of image. With that the window size frame is not changed.

  After you enter the command and select the window, set the new view bounds of image by the "rubber" rectangle.
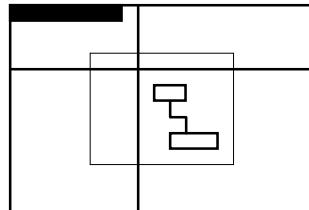
Figure 24 Set the New View Bounds of Image by the "Rubber" Rectangle

After you set the new view bounds of image, the "rubber" rectangle will disappear and the window will be redrawn. Everything enclosed inside the "rubber" rectangle, now will be shown in the window.
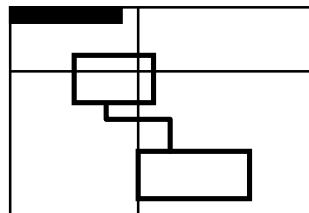
Figure 25 After You Set the New View Bounds of Image, the "Rubber" Rectangle Will Disappear and the Window Will Be Redrawn

You can set the view bounds not only relative to the view in the current window, but relative to the view in any layout window.

- **"?"** (F1) allows you to get the exact co-ordinates of the current point in the window. The values of the point co-ordinates will be displayed in dialog window.

- **"☺"** (SHIFT-ENTER) allows you to enter the point co-ordinates to the layout window with the keyboard.

  After you activate the command and select the window, enter the absolute co-ordinates of point and press the ENTER key.

  If you enter the co-ordinates relative to the last entered values, before corresponding number type the **"d"** symbol. For example, the string

          d110 d-200

  means, that the entering point will be distant from the previous point on 110 units in the X direction (relative to applied co-ordinate system) and on -200 units in the Y direction. Entering values are rounded.

- **"X"** (F2), **"Y"** (F3) turns on/off the view modes, in which the image is reflected in mirror on both X and Y directions. The current state of modes displays in the status line of window.

  The **"X"** symbol in the status line shows, that the image is reflected in mirror on X direction. if the **"X"** symbol is absent in the status line, that means the normal view mode on X direction. The presence or absence of **"Y"** symbol in the status line has the same mean.
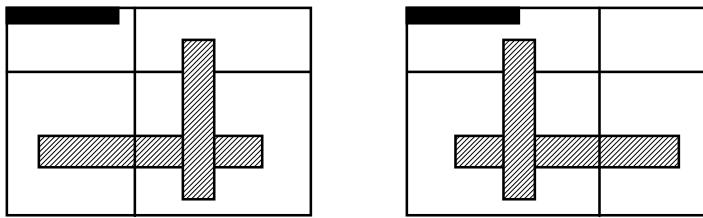
Figure 26 The X Symbol Picking Turns on/off the Mirror Reflecting

- **"H"** specifies the hierarchy depth of viewing cells. After you pick this symbol, the all cells from hierarchy will be shown and the **"H"** symbol will appear in the status line of window. After you pick this symbol once more, only current cell will be shown and the **"H"** symbol will disappear from the status line.

  To set the intermediate value of hierarchy depth of view, use the **"1".."9"** keys while entering command.

Layer table control

In the layer table the information about every layer is represented by the set of symbols (<u>information string</u>), in which following data are contained:

- layer color. It is the same as the information string color.

- fill pattern and line style of layer. The box at the first position of information string shows that.

- layer visibility at the current moment. If the box at first position is visible, then the layer is visible too. In opposite case the layer is invisible.

- layer activity at the current moment. If the small highlighted square is in the box, then this layer is active.

- layer identifiers in the SOURCE (number) format and in the CIF (text string) format.
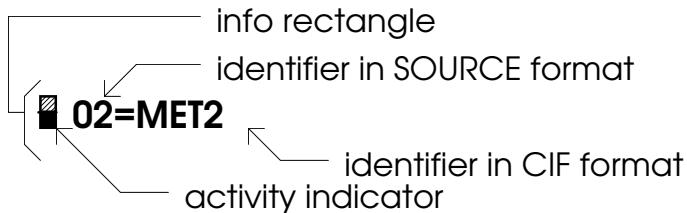
Figure 27 The Information String

A layer in LW can be both visible or invisible. The visible layer also can be an active. To change the state of layer, move the cursor to the information string corresponding the layer and select it (by clicking the left mouse button or pressing ENTER or SPACE keys). After that the layer state will be changed.

The command string of layer window is placed at last line of window. Each symbol from this string corresponds to the some command of layer control. There are two ways to enter the command:

1. **a)**   Pick the command in the command string

    b) Move the cursor to the information string of layer and select it.

   After that the command is executed at once.

2. **a)**   Move the cursor to the information string of layer.

    b) Press the key or the key combination corresponding to necessary command.

   After that the command is executed at once.

The following list contains commands of layer control, its description and symbols, which are corresponded to them:

- **"c"** changes the layer color, increasing its index by one. The color of information string in the layer table is the same as the layer color.

- **"f"** changes the fill pattern of layer, increasing its index by one. The fill pattern of box at the first position of information string is the same as the fill pattern of layer.

- **"l"** changes the line style of layer, increasing its index on the unity. The line style of box at the first position of information string is the same as

the line style of layer, if the fill pattern of layer is transparent.

- **"<", ">"** moves the layer in the layer table to the direction of arrow. With that the order of layers in the layer table is changed.

- **"m"** changes the information, viewed in the layer window. There are four variants:

  - ★ no one of identifiers are not displayed in the layer window

  - ★ only identifier in the SOURCE format (the layer number) is displayed in the layer window

  - ★ only identifier in the CIF format (the layer name) is displayed in the layer window

  - ★ both identifiers are displayed in the layer window

  Note, that in last two variants the size of layer window can be too small for viewing all layer table. To avoid corruption's increase window size.

- **"r"** allows you to load a new layer table into the editor. After you pick this command, enter the name of layer table file and press the ENTER key. If this file reads successfully, the layer window will be redrawn and the information from loaded file will appear in the layer window.

- **"s"** stores the layer table to the file, from which it had been loaded.

Creating objects

To create objects, you first need to define layers object(s) will be on. This is named setting the create layers.

After you pick the menu command, the editor prompts you to execute next actions.

You can delete recently created objects by pressing BACKSPACE key on the keyboard.

Setting the create layer

The state of **"active"-"visible"-"all"** switcher in the CREATE menu specifies layers, on which created primitives will be placed:

- When the state is **"active"**, primitives will be placed only on those layers, which are active at the moment of creation.

- When the state is **"visible"**, primitives will be placed on all visible layers.

- When the state is **"all"**, primitives will be placed on all layers, including invisible.

Set the switcher to the state you want by picking this item until the necessary state will be shown.

Creating basic primitives

To create any of basic primitives, use commands from CREATE menu.

Creating a box

To create a box:

1. Set the **"box"** command of CREATE menu to the active state.

LW prompts you to enter the first corner of box.

2. Enter the first corner of box by clicking mouse or pressing the ENTER key.

LW prompts you to enter the second (opposite) corner of box. When cursor is moving the "rubber" rectangle will appear in the window. This rectangle displays shape of box to be created.

3. Enter the second corner of box by clicking mouse or pressing the ENTER key.

After that the box will be drawn instead of "rubber" rectangle.

Setting the wire parameters

Each wire has two parameters: its width and the value of end margin. To set these parameters:

1. Pick the **"width"** command in the CREATE menu.

LW prompts you to enter values of parameters.

2. Enter values of wire parameters in dialog window and then press the ENTER key.

After that LW sets the wire width and end margin for new wires.

Warnings:

1. The wire width can be only more than zero, In opposite case the editor will display the error when wire will be created.

2. It is recommended to set end margin equal to half of wire width.

Setting the angle mode

Objects with non-rectangular corners (wires, polygons, lines) are created in correspond to current value of angle mode. There are three states of angle mode: 90 degrees, 45 degrees and arbitrary angle (0 to 360 degrees). To set the angle mode state, pick the **"45 s"-"90 s"-"free angle"** switcher until the value you want will be shown.

LW sets the angle mode for new objects.

Creating a wire

To create a wire:

1. Set the **"wire"** command of CREATE menu to the active state.

2. Move the cursor to the position of initial vertex of wire path and enter this point.

   LW draws the initial vertex (square with sides, which are equal to the wire width) and prompts you to enter the next path vertex position.

3. Move the cursor to the position for next vertex on the path.

   When cursor moving the "rubber" line, which starts at the last entering point, follows the cursor.

4. Enter the position of vertex.

   LW draws the part of wire between two last entering vertices and prompts you to enter the next path vertex position.

   To remove last entering point, press the BACKSPACE key.

5. Repeat steps **3-4** until you have created all the vertices for the wire.

6. Finish the vertex entering by pressing the ESC key.

When some operations (entering error, zooming, scrolling) are executed, the "rubber" line disappears, but you can continue to enter vertices. Enter the vertex more once, and the "rubber" line will appear again.

You can't enter the vertex, if the acute angle will be between neighbouring segments of wire. In this case repeat vertex entering correctly.

Creating a line

The order of line creating procedure is the same as when creating wire. Created lines will have the zero width and its end margin will be undefined. Errors are also the same as when creating wire.

Creating a polygon

To create a polygon:

1. Set the **"polygon"** command of CREATE menu to the active state.

2. Set the position of first outline vertex.

LW prompts you to enter the next path vertex position.

3. Move the cursor to position for next vertex on the outline

When cursor moving the "rubber" line, which starts at the last entering point, follows the cursor.

4. Enter the position of outline vertex.

LW draws the part of outline between two last entering vertices and prompts you to enter the position of next vertex.

To remove last entering point, press the BACKSPACE key.

5. Repeat step 2 until you have created all the vertices for the outline.

6. Finish the vertex entering by pressing the ESC key.

After that the polygon will be drawn inside the outline.

When some operations (entering error, zooming, scrolling) are executed, the "rubber" line disappears, but you can

continue to enter vertices. Enter the vertex more once, and the "rubber" line will appear again.

You can't enter the vertex, if the acute angle will be between neighbouring segments of outline. In this case repeat entering vertex correctly.

Instantiating a cell

To instantiate a cell, you first need to specify, what cell you want to instantiate. For that, pick the **"cellname"** command of CREATE menu. The editor prompts you to enter the cell name and shows the name of last instantiated cell:

Filename *cell_name*

Enter the cell name you want and press the ENTER key.

After that the editor memorises this name of cell and when placing the instance it will place the instance to cell with entering name.

To place the instance, set the **"cell"** command to the active state. LW prompts you to place the instance of cell with early entered name:

Place cell *cell_name*

Enter the position of instance base point by clicking the left mouse button or pressing the ENTER (SPACE) key. The base point of instance has (0,0) co-ordinate values relative to instantiated cell co-ordinate system.

You also can place the instance using the commands of CELL menu. If you use commands of this menu, first you must specify the name of instantiated cell by picking the **"cellname"** command and entering the cell name you need.

After that, place the instance by activating the **"insert"** command and setting the position of the instance. Using these menu commands you can rotate the contents of the instantiated cell, zoom it and reflect it in mirror. Use the **"Wmirror"**, **"Xmirror"**, **"Ymirror"** flags to reflect contents. To zoom the cell's contents, activate the **"scale"** flag and for rotating use the **"0 s"-"90 s"-"180 s"-"270 s"** switcher. In detail it explains in the *Section 5. Menu Dictionary. CELL menu.*

Creating an array of basic primitives

You can create an array of any basic primitives. For that, before creating the object, on which the array will be based, set the **"array"** item to the active state. After you

pick this item, LW prompts you to specify the array parameters:

Array n(nx,ny) m(mx,my) *num_n n_offset_on_x*
n_offset_on_y
num_m m_offset_on_x m_offset_on_y

where

num_n and *num_m* specifies the number of elements on M and N directions of array spreading

n_offset_on_x, *n_offset_on_y,* *m_offset_on_x,* *m_offset_on_y* are horizontal and vertical offsets between neighbouring elements in the array on both directions.

Enter the array parameters and press the ENTER key when entering finished.

After that when you pick the command, which creates one of basic primitives, and execute it the array of the same primitives will be displayed in the window.

To turn off the mode, when arrays are creates, set the **"array"** flag to passive state.

Creating arc primitives

Commands to create arc primitives are placed in the ARCS menu. Before activating the creating command you can specify the number of arc segments and width and end margin for ring or arc.

Settings the number of arc segments

Now, in LW you can set only two values of this parameter: 8 and 56. To set the number of arc segments, set the **"8 sides"-"56 sides"** switcher to the state you want.

LW sets the number of segments for new arc primitives.

Setting the width and end margin for ring and arc

To set the width and the end margin of arc primitives (ring and arc), pick the **"width"** command of ARCS menu. The editor prompts you to set the values of these parameters. Enter the values you want and then press the ENTER key

LW sets the width and end margin for new arc primitives.

Creating a circle

To create a circle:

1. Set the **"circle"** command of ARCS menu to the active state.

2. Set the position of circle centre by clicking the left mouse button or pressing the ENTER (SPACE) key.

   LW prompts you to enter the point on desired outline of circle.

3. Move the cursor to the point on desired outline of circle.

   When the cursor is moved, the "rubber" line, which is started in the entered centre of circle, follows the cursor. The "rubber" line shows the value of circle radius, which will be set, if you enter the current point immediately.

4. Set the position of the point, which will be placed on the outline of circle.

After that the circle will be drawn.

Creating a ring

To create a ring:

1. Set the **"ring"** command of ARCS menu to the active state.

2. Set the position of ring centre by clicking the left mouse button or pressing the ENTER (SPACE) key.

   LW prompts you to enter the point on desired outline of ring.

3. Move the cursor to the point on desired outline of ring.

   When the cursor is moved, the "rubber" line, which is started in the entered centre of ring, follows the cursor. The "rubber" line shows the value of out ring radius, which will be set, if you enter the current point immediately.

4. Set the position of the point, which will be place on the outline of ring.

After that the ring will be drawn.

The ring will be not drawn, if the its width is more than its out radius. In this case repeat the entering the point placed on the ring outline.

Creating an arc

To create an arc:

1. Set the **"arc"** command of ARCS menu to the active state.

2. Set the position of arc centre by clicking the left mouse button or pressing the ENTER (SPACE) key.

   LW prompts you to enter the point on desired outline of arc.

3. Move the cursor to the point on desired outline of arc.

   When the cursor is moved, the "rubber" line, which is started in the entered centre of arc, follows the cursor. The "rubber" line shows the value of out arc radius, which will be set, if you enter the current point immediately.

4. Set the position of the point, which will be place on the outline of arc.

   After that the ring with the same radius will be drawn and LW prompts you to enter the start angle of arc.

5. Set the start angle of arc.

   LW prompts you to enter the end angle of arc.

6. Set the end angle of arc.

After that the arc will be drawn instead of ring. Start and point margins of arc will be specified by entered start and end angles.

Warning: when creating an arc some bugs sometimes appear and arc does not drawn.

Creating a text

You can create a text with commands in the CELL menu. Before creating the text you can change current layout font for text and its height. It discussed in the chapters *Setting the layout font* and *Setting the text height.*

You can also rotate created text string and reflect it in mirror while setting the corresponding flags in the CELL

menu to the active state. It discussed in the chapter *Flags using while text creating*.

Setting the layout font

To set a layout font for text, pick the **"font"** command of the CELL menu. The editor prompts you to enter the font name:

Font *cur_font*

where *cur_font* is the current font. If the current font satisfies you, press the ENTER key, in opposite case enter the font name you want and press the ENTER key.

LW will be create text string using selected font.

Standard LW includes two layout fonts:

- thin (linear) font named **kd** intended to use in documentation

- thick font named **pr3** intended to use in layouts (VLSI, PCB, etc.)

You can create your own layout font. In detail it is discussed in the *Section 7. Miscellanea. The creation of layout fonts*.

Setting the text height

To set the height of created text string, pick the **"height"** command of the CELL menu and enter the text height you want. By default the text height is equal to 100.

It is recommended to set pr3 text height value snapped to 20 in order to avoid transformation errors (for example, 1720, 380, 20).

Setting the text string

To set a text string into the cell layout:

1. Pick the **"text"** command of the CELL menu.

2. Enter the position of text string. The entered point will be in the left and lower corner of string.

LW prompts you to select the layer on which the text string will be placed.

3. Select the layer you want in the layer table.

LW prompts you to enter the text string you want to set into the layout.

4. Enter the text string in dialog window. Finish entering by pressing the ENTER key.

After that the text string will appear in the current window on the selected layer.

Creating mirrored and rotated text

When you set the text string, there are some flags in the CELL menu, which can be used for rotating and mirror reflecting created text. These flags are set before executing the **"text"** command. These flags are:

- **"Xmirror"** flag is used for mirror reflecting the text string to the X direction

- **"Ymirror"** flag is used for mirror reflecting the text string to the Y direction

- **"0 s"-"90 s"-"180 s"-"270 s"** flag switcher is used for rotating text string relative to the base point on the selected angle

Editing objects

There are three edit menu variants in LW: CHANGE menu, TRANSFOR-MATION menu and SPECIAL TRANSFORMATION menu. To edit objects, you first need to set one of them. To select objects, you can use both the commands from the SELECT menu and the commands in the current edit menu.

You can cancel the just executed edit command by pressing the BACKSPACE key on the keyboard.

Selecting objects

You can select objects by commands of the SELECT menu and by commands in the current edit menu as well. The difference between these ways is that in last way you select objects just after the edit command is activated and after command executing the selected objects becomes non-active. If you select objects with the help of SELECT menu commands, the selected object keeps their state after executing command.

In both ways of selection there are two flag switchers, which are very important when object selecting. These flags are in all edit menus and also in the SELECT menu. They are:

- the   **"active"-"visible"-"all"**   switcher,   which
  specifies layers, on which objects can be selected

- the   **"point"-"area"-"global"-"selected"**   switcher,
  which defines the select mode

In the SELECT menu the last switcher does not contain the
"selected" state.

The   states   of   **"active"-"visible"-"all"**   switcher   have
following mean:

- In the **"active"** state only active layers can contain
  selected primitives.

- In the **"visible"** state only visible layers can
  contain selected primitives.

- In the **"all"** state selected objects may be placed on
  any of all layers.

The select mode switcher defines way, by which you can
select objects:

- **selection by point.** In this way you select the object
  by clicking the left mouse button or pressing the
  ENTER (SPACE) key over it.

- **selection by area.** In this way you select objects by
  "rubber" rectangle. All objects, intersected with
  this rectangle or surrounded by it, will be selected.

- **global selection.** In this case you select all
  objects, which can be selected.

- the **"selected"** state of switcher specifies that the
  objects are already selected (or will be selected)
  with the commands of SELECT menu.

To select objects by the SELECT menu commands:

1. Switch to the SELECT menu and activate the **"select"**
   command.

2. Set the layer switcher and select mode switcher to
   the states you need.

3. Select object(s) you want by way you set in the
   select mode switcher.

   Selected objects will be highlighted (usually by the
   white color).

> **4.** Just you have finished the selecting, return to the edit menu and set the select mode switcher to the **"selected"** state.

You can select objects by this way both before edit command executing and when the edit command is activate.

In the SELECT menu you can also unselect objects or invert selecting by activating the **"unselect"** and **"invert"** commands correspondingly. After activating the command you must select window, in which this command will be executed.

To select objects placing in the current edit menu, you must have already activated edit command. Then you must:

> **1.** Set the layer switcher and select mode switcher to the states you need.

> **2.** Select object (objects) you want by way you set in the select mode switcher.

After that the edit command will be executed over the selected objects and then objects will become non-active.

Transforming objects

The commands of object transforming are placed in two menu variants: in the TRANSFORMATION menu and in the SPECIAL TRANSFORMATION menu.

These menu variants contain almost the same edit commands and flags. The difference between them is that in the SPECIAL TRANSFORMATION menu you can also rotate objects, reflect it in mirror, zoom them, etc.

Deleting objects

The deleting command is contained in each of edit menus. To delete already selected objects:

> **1.** Set the **"delete"** command to the active state. The **"point"-"area"-"global"-"selected"** switcher must be in the **"selected"** state.

> **2.** Select window, in which selected objects are placed.

Selected objects will disappear from the window.

To delete yet not selected objects:

> **1.** Set the **"delete"** command to the active state.

> **2.** Set the layer switcher and select mode switcher to states you need.

3. Select objects you want.

After that objects, which you select, will disappear from the window.

Changing the size of edited objects

When objects are copied or moved, you can change it size. To change the size of edited objects, set the **"resize"** flag to the active state.

LW prompts you to enter the change size you want:

Resize   *def_size*

where *def_size* is the last specified change size (by default it equal to zero). Enter the value you need instead of the *def_size* and press the ENTER key. If the entered value is positive, the edited objects will be increased. In opposite case they will be decreased. In all cases correction is performed in direction, normal to primitive's side.

LW sets the change size to the new edited objects. If you set the **"resize"** flag passive, the size of edited object will not be changed.

Copying objects

The **"copy"** command is contained in both transforming menu variants. To copy already selected objects:

1. Set the **"copy"** command to the active state. The **"point"-"area"-"global"-"selected"** switcher must be in the **"selected"** state.

2. Select the base point of copied objects.

LW prompts you to enter the new position of base point.

After that when the cursor is moved, the "rubber" line will appear in the window. It shows the direction and distance between selected objects and objects, which will be create.

3. Select the new position of base point.

After that selected objects will be copied to the new place.

To copy yet not selected objects:

1. Set the **"copy"** command to the active state.

2. Set the layer switcher and select mode switcher to states you need.

3. Select objects you want.

LW prompts you to select the base point of selected objects.

4. Select the base point of selected objects.

LW prompts you to select the new position of the base point.

After that when the cursor is moved, the "rubber" line will appear in the window. It shows the direction and distance between selected objects and objects, which will be create.

5. Select the new position of the base point.

After that objects, which you select, will be copied from the window.

You can copy objects not only in current window, but also to other windows.

Moving objects

The **"move"** command is contained in both transforming menu variants. To move already selected objects:

1. Set the **"move"** command to the active state. The **"point"-"area"-"global"-"selected"** switcher must have the **"selected"** value.

2. Select the base point of moved objects.

LW prompts you to enter the new position of base point.

After that when the cursor is moved, the "rubber" line will appear in the window. It shows the direction and distance on which the selected object will be moved.

3. Select the new position of base point.

After that selected objects will be moved to the new place.

To move yet not selected objects:

1. Set the **"move"** command to the active state.

2. Set the layer switcher and select mode switcher to states you need.

3. Select objects you want.

   LW prompts you to select the base point of selected objects.

4. Select the base point of selected objects.

   LW prompts you to select the new position of the base point.

   After that when the cursor is moved, the "rubber" line will appear in the window. It shows the direction and distance on which the selected object will be moved.

5. Select the new position of the base point.

After that objects, which you select, will be moved from the window.

You can move objects not only in current window, but also to other windows.

Setting the target layer

When you transform objects, you can place them both in the same layer, in which they were placed, and in another layer you want. To set a target layer, set the **"same layer"** flag to the active state. This flag is contained in both menu variants.

After that when the edit command will be execute, the editor prompts you to select target layer. Select target layer in the layer window.

After that the edit command will be executed and objects will be placed in the target layer.

Special transformating modes

You can make the **"copy"** and **"move"** commands more powerful by activating the various flags in the SPECIAL TRANSFORMATION menu. With these flags you can rotate objects, reflect it in mirror, zoom it. These flags are:

- **"Xmirror"**, **"Ymirror"** flags allow you to reflect the selected objects in mirror on both directions.

- **"0 s"-"90 s"-"180 s"-"270 s"** switcher specifies angle, on which the edited objects will be rotated when the edit command is finished.

- **"scale"** flag allows you to specify the zooming coefficient for edited objects. This coefficient specifies by the proper fraction. For example, to

enter the coefficient, which is equal to 0.01, you
need to enter the following string in dialog window:

>   1 100

- **"array"** flag specifies, how the selected array is
  represented for editing:

  - ★ when this flag is active, the array is represented
    as one object and it will be edited as one object.

  - ★ when this flag is passive, the array elements are
    single objects and you will edit the elements of
    array, but not the array as one object.

You can activate these flags before the edit command
executing.

Deforming objects

In LW you can deform objects, i.e. change its shape. To
deform objects, you must use the **"deform"** command. This
command is contained only in the TRANSFORMATION menu. To
deform object:

1. Set the **"deform"** command to the active state.

2. Set the **"point"-"area"-"global"-"selected"** switcher
   to the **"area"** state.

3. Select vertices of objects you want to deform by
   "rubber" rectangle.

   LW prompts you to enter the base point.

4. Select the base point.

   LW prompts you to enter the new placement of the base
   point.

   After that when cursor is moved, the "rubber" line
   will appear. It shows the direction and distance, on
   which the selected vertices will be moved.

5. Select the new placement of the base point.

After that objects, in which you select the vertices, will
be redrawn and edited vertices will be in the new
positions.

Warning: when deforming the editor checks deformed
primitives on the acute angles and nonzero width and height
of boxes. Thus, if you deform objects non-correctly, they
will net be redrawn. In this case repeat all steps of
command executing again.

You can specify the exact moving of vertices concurrently to the orthogonal axes by activating the **"ortogonal"** flag.

Modifying objects

This chapter contains description of commands, which are more specific than previously describing commands. These commands are placed in the CHANGE menu.

Flattening an instance

To flatten already selected instances, set the **"uncell"** command to the active state. The **"point"-"area"-"global"-"selected"** switcher must be in the **"selected"** state. Then you must select window, in which the selected instances are placed.

After that selected instances will be flatten.

To flatten yet not selected instances:

1. Set the **"uncell"** command to the active state.

2. Set the layer switcher and select mode switcher to states you need.

3. Select instances you want to flatten.

After that instances, which you select, will be flatten.

Unfolding an array

The array unfolding means, that the all elements of array will be represent as single elements after this command executing. To unfold the already selected arrays:

1. Set the **"unarray"** command to the active state. The **"point"-"area"-"global"-"selected"** switcher must be in the **"selected"** state.

2. Select window, in which selected arrays are placed.

After that these arrays will be redrawn and each element will be already represented as single object.

To unfold yet not selected arrays:

1. Set the **"unarray"** command to the active state.

2. Set the layer switcher and select mode switcher to states you need.

3. Select arrays you want to unfold.

After that these arrays will be redrawn and each element will be already represented as single object.

Changing an object property

To change an object property, you first need to select objects. Then you must pick the **"property"** command.

LW prompts you to enter the new property of selected objects. Type new property string in dialog window and press the ENTER key.

After that objects you select will be have a new property string.

Changing array parameters

To change parameters of array, you first need to select it. Then pick the **"array"** command. The editor prompts you to enter the parameters of array and displays the current values. Enter the new values of array parameters instead of current values, if current values are not satisfies you, and press the ENTER key.

After that the selected array will be redrawn with the stock of just entered parameters.

Exploding an object

You can explode a wire or a line in the breakpoints of object path. For that:

1. Set the **"p.divide"** command to the active state and the **"point"-"area"-"global"-"selected"** switcher to the **"area"** state.

2. Outline the breakpoint of object, in which you want to explode it.

After that the exploded object will be redrawn and already there will be two objects instead of one.

Design rule checking

LW provides design rule checking (DRC) for project layout. In LW there are several rules, by which you can check your layout. You may check layout both interactively by one rule and in command mode through rule file which can contain several rules.

You can check layout by rules both automatically when object creating or editing (in create or edit menus correspondingly) and by executing **"check"** command from

CHECK menu. Below the last way of checking is discussed. The first way is discussed in corresponding menu descriptions in *Section 5 Menu Dictionary*.

Check rules

In LW there are 6 rules for design rule checking:

- size rule

- distance rule

- overlap rule

- inclusion rule

- gate extension rule

- gate distance rule

and one rule for electrical circuit checking:

- electrical distance rule

All rules are described below. To set any of them, two layers must be selected (for size rule - one layer). The first layer is the target layer. Duaring rule checking primitives placed on the target layer are checked with respect to primitives placed on second selected layer. Target and second layers may be coincide.

There are two qualitative criterions of rule checking (minimum and maximum) and numerical criterion. If the maximum criterion is set, it means that all checked values must be not more than numerical criterion. When minimum criterion is set, all checked values must be not less than numerical criterion.

Size rule

This rule is used for checking sizes of objects. In dependence of qualitative criterion (minimum or maximum) two object sizes can be checked. For example, in the polygon, which shown in the Figure 28, two sizes can be checked in dependence of criterion.
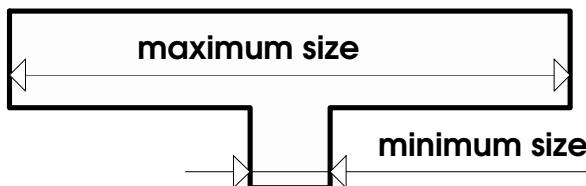
Figure 28 The Size Checking

Distance rule

This rule is used for checking distances between primitives, which placed on target layer, and primitives placed on the second layer. Figure 29 shows two primitives and checked distance between them. The primitive placed on target layer is outlined by bold line.
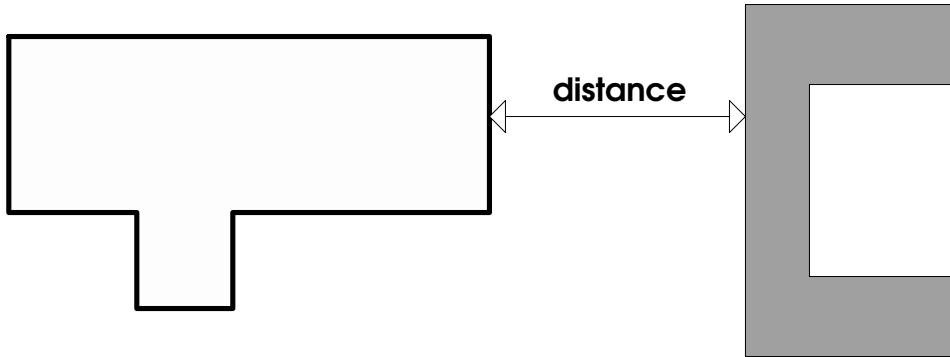
Figure 29 The Distance Checking

Electrical distance rule

This rule is used for electrical circuit checking on short current, exactly on minimum distance between circuit with different potentials.

For checking electrical circuits you must set properties of elements by following way:

- The property of electrical circuit (primitive, which is the circuit) is specified as

    5 *circuit_identifier*

    where *circuit_identifier* is the text string of arbitrary contents. In Figure 30 you can see an example of specifying properties to the circuit. The property string **"5 VT1 n1 n2 n3"** is assigned to instance of cell named a.cif.
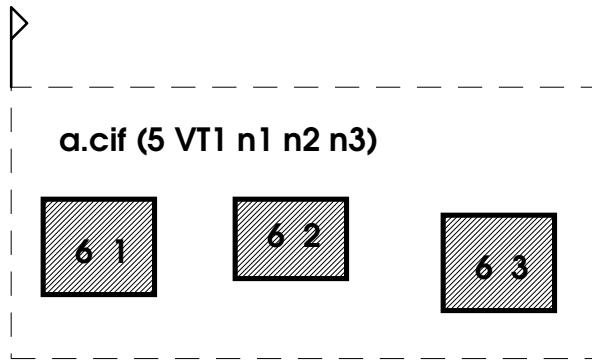
Figure 30 Example of Specifying Properties

- The property of instance is set in following format:

    5 *elem_identifier output_identifier*_1 *...*

    *output_identifier*_N

  where *elem_identifier* is the own instance identifier
  and *output_identifier*_1 *... output_identifier*_N are
  output identifi-ers (text string ended by index of
  output).

- Properties   of   instance   outports   have   following
  format:

    6 *output_number*

  where *output_number* is the ordinal number of instance
  outport. Figure 30 shows the examples of properties
  for   instance   and   it   outports.   Outports   of
  instantiated cell contain properties **"6 2"**, **"6 2"**, **"6
  3"** correspondingly.
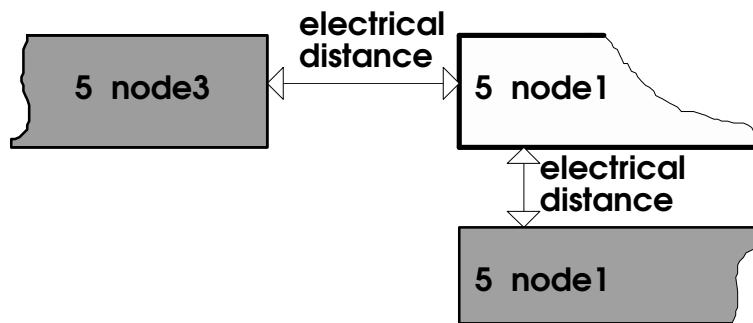
Figure 31 The Electrical Distance Checking

The electrical rule checking is executed by following way:
the editor check on distance only primitves, which have
different identifiers. In the Figure 31 you can see three
primitives. For primitive outline by bold line only one

distance (between it and primitive with property 5 node3) will be checked. Another distance will not be checked.

Overlap rule

This rule is used for checking the clearance between overlapping and overlapped objects. The checking is performed with respect to overlapping primitives. Figure 32 shows example of this placement and checked distance.
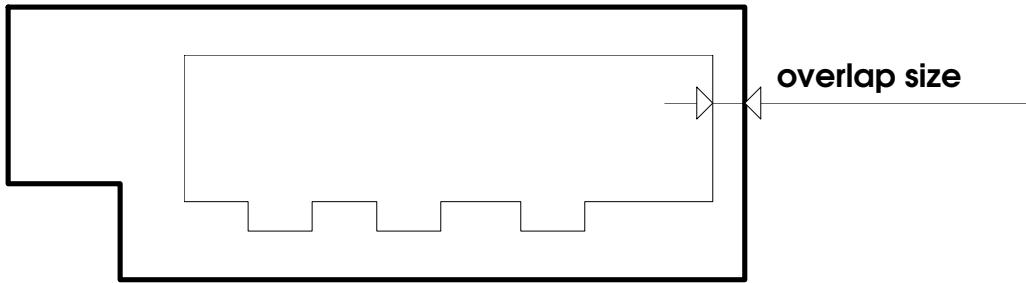
Figure 32 The Overlap Checking

Inclusion rule

This rule is also used for checking the clearance between overlapping and overlapped objects. But in this rule the checking is performed with respect to overlapped primitives. Figure 33 shows example of that placement and checked distance.
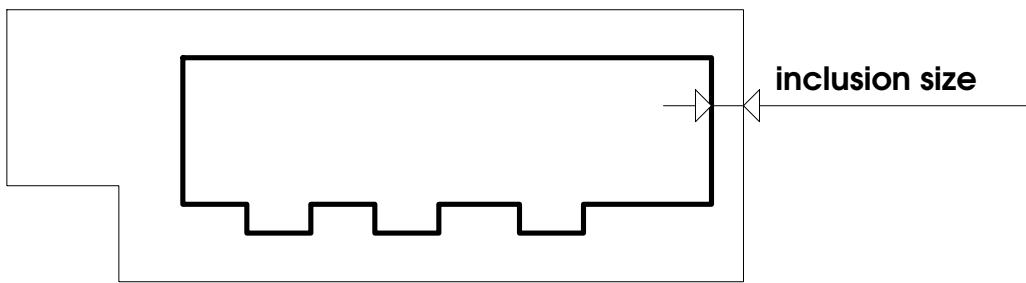
Figure 33 The Inclusion Checking

Gate extension rule and gate distance rule

These rules are used when checking layout of MOS or CMOS schemes. They checks the distance between edge of gate and nearest edge of channel called gate extension and the distance between passive part of gate and nearest edge of channel. Figure 34 shows the piece of MOS layout and checked distances. The checking is performed with respect to gates.
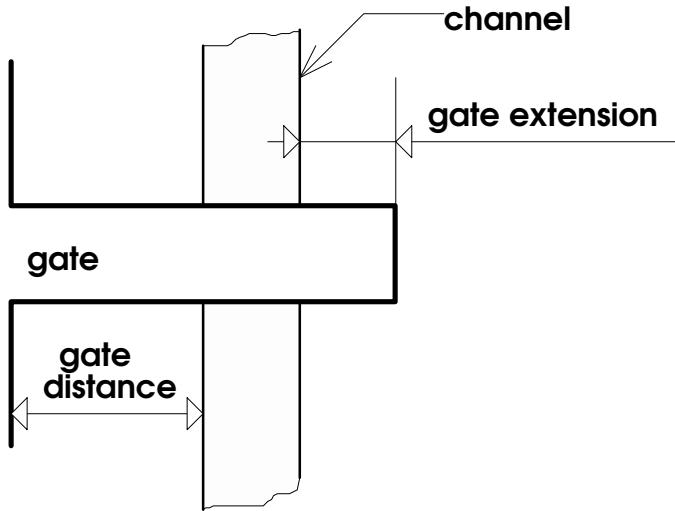
Figure 34 The Gate Extension and Gate Distance Checking

One rule checking

To check layout by some rule, you must set the rule, its qualitative criterion (minimum or maximum) and enter the value of numerical criterion. It achieves by executing commands from CHECK menu.

For example, you need to check your layout on distance between primitives. The distance must be not more than 50 units. To check layout by this rule:

1. Switch to CHECK menu.

2. Select criterion by setting the **"min"-"max"** switcher to the state you need. The **"min"** state means, that the distance must be not less than needed value.

3. Select numerical criterion by picking the **"size"** command. The editor prompts you to enter the value of criterion.

4. Type the value of criterion and press the ENTER key.

5. Select the rule by setting the **"rules"-"size"- "distance"-"el.list"-"overlap"-"inclusion"-"gate ext"-"gate dis"** flag switcher to state you need (except **"rules"** state).

6. Select way of selecting the checked area by setting the **"global"-"area"** switcher to needed state. In **"global"** state whole cell layout will be checked. In opposite case you must select the checked piece of layout by "rubber" rectangle.

7. Select checked cells by setting **"current"-"hierarchy"** switcher to needed state. In **"current"** state only current cell will be checked. In second case layout of all cells from cell hierarchy will be checked.

8. Activate the **"check"** command and select layout to check.

9. Select the target layer and second layer, if it is needed.

After these actions the editor checks selected piece of layout by current rule and displays results of checking in dialog window.

There are some serve actions for checking:

- setting the error layer, on which error primitives will be outlined

- setting the error file, where error primitives will be stored

This is achieved by picking menu items **"err.layer"** and **"err.file"** correspondingly.

File based checking

Before command checking of layout you must have the design rule checking file (DRC file), in which all rules are described.

To check layout by rules from DRC file:

1. Switch to CHECK menu.

2. Set the **"rules"-"size"-"distance"-"el.list"-"overlap"-"inclusion"-"gate ext"-"gate dis"** flag switcher to the **"rules"** state.

3. Pick the **"rules file"** command and enter the DRC file name.

4. Select way of selecting the checked area by setting the **"global"-"area"** switcher to needed state. In **"global"** state whole cell layout will be checked. In opposite case you must select the checked piece of layout by "rubber" rectangle.

5. Select checked cells by setting **"current"-"hierarchy"** switcher to needed state. In **"current"** state only current cell will be checked. In another case layout of all cells from cell hierarchy will be checked.

6. Activate the **"check"** command and select checked
layout of layout.

After these actions the editor checks selected piece of
layout by rules contained in the DRC file and displays
results of checking in dialog window.

In order to interrupt checking after completing each rule
activate **"stop"** flag.

Creating DRC file

Writing DRC file you must keep in mind the following rules:

- All file consist of two parts: title and set of
  rules.

- The file title contains only the format name ("CIF"
  or "SOURCE").

- In the next lines all rules are placed in the file –
  each rule is placed on one line.

The format of rule description is follows:

> *identifier = tar_layer rule_symbol_1 num_criterion
> qua_criterion rule_symbol_2 sec_layer > err_layer
> prim_classes*

where

> *identifier* is text string with arbitrary contents up
> to 15 symbols,
>
> =, > are delimiters between fields,
>
> *tar_layer* is checked layer,
>
> *rule_symbol_1, rule_symbol_2* are symbols that specify
> the rule (they are described below),
>
> *num_criterion* is numerical criterion of rule,
>
> *qua_criterion* is qualitative criterion of rule (**min**
> or **max**),
>
> *sec_layer* is the second selected layer, relatively to
> which target layer is checked,
>
> *err_layer* is the layer, on which error primitives
> will be outlined. If description of this layer is
> missed, by default the CALL-00 layer is set as error
> layer
>
> *prim_classes* are classes of checked primitives (it is
> described below).

The following list contains rules, corresponding symbols and their description:

- [] is the object size rule

- () is the intersection rule

-)(is the distance rule

-][is the electrical distance rule (the checking on short current)

-)) is the inclusion rule

- ((is the overlap rule

- [(is the gate extension rule

-)] is the rule, which checks the distance between edge of channel and the gate edge

-)[is the rule, which checks the distance between edge of channel and the passive edge of gate

-](is the gate distance rule

The checked primitive classes are specified by text string without spaces. Each symbol of this string is corresponded to one of primitive's classes. These symbols are:

- **B** specifies boxes

- **P** specifies polygons

- **W** specifies wires

- **L** specifies lines

The following list is the small example of DRC file:

```
CIF
r1 = al1 [ 80min ]
a34 = al2 ) 40min ( al1 > call W
```

The default DRC file listing is contained in *Appendix D. Default DRC file.*

Section 5
Menu Dictionary

Organization of section

This section is organised by following way:

- The whole section is divided on chapters, each of which contains information about one menu variant.

The description of each menu variant is composed in following order:

- Destination of this menu commands

- Description of menu switchers

- Description of menu commands

- Description of menu flags and flag switchers

The FILE menu is the exception. In description of this menu variant after each command description the corresponding flags are described.

MAIN menu

The MAIN menu is the main of menus. It contains switchers to the other menu variants and some auxiliary commands.

```
LW 1.27
quit
file I/O
plot
cell
create
transfor
change
select
check
setup

dbase  ls
memory
director
OS  cmd
```
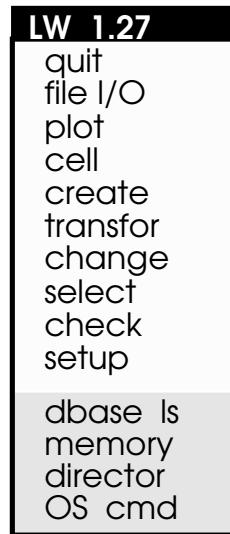
Figure 35 The MAIN Menu

Description of menu switches

The MAIN menu contains following menu switches:

- **"file I/O"** is the switch to the FILE menu, where you can open and store cells.

- **"plot"** is the switch to the PLOT menu, where you can print your layout on the printer or plotter.

- **"cell"** is the switch to the CELL menu, where you can manage cells and create the text.

- **"create"** is the switch to the CREATE menu, where you can create primitives.

- **"transform"** is the switch to the TRANSFORMATION menu, where you can transform objects (delete it, move, copy, deform).

- **"change"** is the switch to the CHANGE menu, where you can perform specific operations over the objects (flatten the cell, unfold the array, etc.).

- **"select"** is the switch to the SELECT menu, where you can select objects for editing.

- **"check"** is the switch to the CHECK menu, where you can execute the design rule checking.

- **"setup"** is the switch to the SETUP menu, where you can set the window configuration and database parameters.

Description of commands

The "quit" command

This command is used for quitting from the editor. After you pick it, the editor checks cells for updating. If the some cells will be not updated, the editor displays the list of non-updated cells and asks you:

    Really quit?(y/[n])

To cancel the quit, enter the **"n"** character (or press ENTER at once). In opposite case enter the **"y"** character and press the ENTER key. After that the edit session is finished.

The "dbase ls" command

When you pick this command the editor displays the list of cells, which are placed in the editor's database.

The "memory" command

This menu command is used for displaying the size of memory busied by the editor. When you pick this command, the editor prints in dialog window:

    Allocated *mem_size* kbyte(s)

where *mem_size* is the size of busied memory (in kilobyte units).

The "directory" command

The **"directory"** command is used for displaying the directory list. After you pick this command, the editor prompts you to enter the directory path:

    dir

If you want to see the list of current directory, press the ENTER key at once. Else, enter path to the directory you want.

After you press the ENTER key, the editor prints the directory list in dialog window.

The "OS cmd" command

This command is used for executing commands of your OS. After you pick this command, the editor prompts you to enter the OS command in dialog window.

After you enter the OS command and press the ENTER key, this commands will be executed and the result of it executing will be displayed in dialog window.

FILE menu

Commands from this menu variant are used for opening and storing cells. It contains one menu switcher (**"back")**, two base commands (**"read"** and **"write"**), several auxiliary commands and many-many flags. Each of flags is used for both **"read"** command or **"write"** command. Only **"CIF"-"SOURCE"-"CIF 2.0"** flag is exception. It is used both while the cell loading and while the cell storing.

```
┌─────────────────┐
│ FILE I/O        │
├─────────────────┤
│ back            │
│ dbase ls        │
│ memory          │
│ director        │
│ OS cmd          │
├─────────────────┤
│ err. lim.       │
│ levels          │
│ win. fram       │
│ path            │
├─────────────────┤
│ read            │
├─────────────────┤
│ all             │
│ global          │
│ current         │
│ scale           │
│ box >pol        │
│ wire>pol        │
│ confirm         │
├─────────────────┤
│ write           │
├─────────────────┤
│ CIF             │
└─────────────────┘
```

Figure 36 The FILE Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands and flags

In this chapter only two commands (**"read"** and **"write"**) are described. Other (auxiliary) commands are described above in the MAIN menu description.

The "read" command

This command is used for opening the cell. After activating this command you need to select window, in which the opened cell will be shown. Then you must enter the file name of opened cell.

Read flags

When opening the cell you can set some flags in the FILE menu for setting the modes you need.

- **"err.lim"** flag in the active state specifies the number of read errors, after detecting of which the editor breaks the reading and asks you - continue the reading or not. If this flag is non-active, the editor not breaks the reading until the end of file will be detected.

- **"levels"** flag in the active state specifies the hierarchy depth on which the cell reading will be executed. If this flag is non-active, all cell hierarchy will be loaded.

- **"win.frame"** flag in the active state specifies, that the whole loaded cell will be shown in the window. If this flag is passive, the image view bounds remains by previous.

- **"path"** flag is used for specifying the directory list, in which the editor will search the opened cells.

- **"CIF"-"SOURCE"-"CIF 2.0"** flag switcher is used for specifying the file format of loaded cell.

The "write" command

This command is used for storing the cell. After activating the command you must select window with stored cell and enter the file name of stored cell.

Write flags

When storing the cell you can set some flags in the FILE menu for setting the modes you need.

- **"active"-"visible"-"all"** switcher is used for specifying layers, on which the stored primitives are placed.

- **"point"-"area"-"selected"-"global"** switcher specifies the select mode, with the help of which stored primitives will be selected.

- **"current"-"flatten"-"hierarchy"** switcher specifies the file structure of cell hierarchy. In first case only current cell will be stored. In next case all cell hierarchy will

be stored in one file. And in last case all cells from hierarchy will be stored in its own files.

- **"scale"** flag specifies the zooming coefficient of primitive's co-ordinate in the stored cell. When this flag is activated, you must enter two numbers: numerator and denominator of proper fraction, which is the zooming coefficient. If this flag is passive, primitive's co-ordinates are saved to the file with normal scale (1:1).

- **"box>pol"** and **"wire>pol"** flags are used when the transformation of box or wire to the polygon is needed.

- **"confirm"** flag in the active state provides, that when the cell is stored, the editor asks you the name of stored cell and overwrite confirming.

- **"CIF"-"SOURCE"-"CIF 2.0"** flag switcher is used for specifying the file format of stored cell.

PLOT menu

With commands of this menu you can plot or print layout of any shown cell. The current version of LW provides printing in HP-GL language on the LaserJetIII printer and plotting on the following plotters: HP7475A, HP7595, SPL-880, SPL-450.

```
 PLOT
 back

 visible
 current
 0 s
 proport
 A4
 pen auto
 HP7475A

 file/por
 pens
 plot
 dbase ls
 director
 OS cmd
```
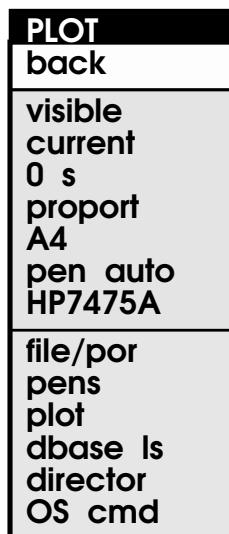
Figure 37 The PLOT Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

In this chapter only three plot commands (**"file/port"**, **"pens"** and **"plot"**) are described. Other auxiliary commands are described above in the MAIN menu description.

- **"file/port"** is used for specifying port or file, what way the output information will be sent. After you pick this menu command, the editor prompts you to enter the name of port or file.

- **"pens"** is used for specifying the number of drawing devices (pens for plotter). After you pick this menu command, the editor prompts you to enter the number of devices. By default it is equal to 4.

- **"plot"** is used for making hardcopy. After setting this command to active state you must select plotted area in the window by "rubber" rectangle. Then output information will

be sent to the port of hardcopy device or to the file (in dependence of your choosing).

Description of flags

- **"active"-"visible"-"all"** flag switcher is used for specifying set of layers, which contains plotted primitives.

- **"current"-"flatten"** flag switcher is used for specifying set of cells, the layout from which will be plotted. If the switcher is in the **"current"** state, only current cell layout will be plotted. In opposite case all cell hierarchy will be plotted.

- **"0 s"-"90 s"** flag switcher is used for specifying the orientation of paper sheet. The **"0 s"** state is corresponded to portrait orientation, and another state is corresponded to landscape orientation.

- **"proport."-"fixed"** flag switcher is used for specifying the bounds of plotted area. If the **"proport."** state is set, the plotted area is the roughly same as you outline in the window.

  In opposite case when plotting the editor prompts you to enter the fixed scale of plotted area. Then the layout will be plotted so, that the bounds of plotted area will be distant from left and upper corner of selected area taking into account the scale coefficient.

- **"A0"-"A1"-"A2"-"A3"-A4"** flag switcher is used for specifying size of paper sheet.

- **"pen auto"-"pen man"** flag switcher is used for specifying the way of selecting plot devices (pens). In **"pen auto"** state all devices are specified automatically, and in opposite case the editor will prompt you to enter the device number for plotting each layer.

- **"HP7475A"-"HP7595"-"LaserJetIII"-"SPL-880"-"SPL-450"** flag switcher is used for specifying printer or plotter.

CREATE menu

This menu variant contains commands for creating basic primitives.
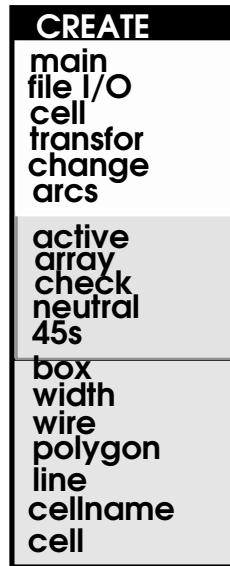
Figure 38 The CREATE Menu

Description of menu switches

The CREATE menu contains following switchers to the other menu variants:

- **"back"** switcher is used for returning to the previous menu variant.

- **"file I/O"** switcher is used for switching to the FILE menu.

- **"cell"** switcher is used for switching to the CELL menu.

- **"create"** is the switch to the CREATE menu.

- **"transform"** switcher is used for switching to the TRANSFORMATION menu.

- **"change"** switcher is used for switching to the CHANGE menu.

- **"arcs"** switcher is used for switching to the ARCS menu, where you can create arc primitives.

Description of commands

- **"box"** is used for box creating.

- **"width"** is used for setting width and value of end margin for path. After you pick this command, the editor prompts you to enter these values in dialog window.

- **"wire"** is used for wire creating.

- **"polygon"** is used for polygon creating.

- **"line"** is used for line creating.

- **"cellname"** is used for specifying the name of instantiated cell. After you pick this command, the editor prompts you to enter the name of instantiated cell.

- **"cell"** is used for instantiating the cell, the name of which is specified when **"cellname"** command picking. By default the cell name is new.l.

Description of flags

- **"active"-"visible"-"selected"** flag switcher is used for setting the set of layer, on which created primitives will be placed.

- **"array"** flag in active state specifies, that the array of primitives is created instead of single primitive after the creating command is executed. After you activate this flag, the editor prompts you to enter the array parameters, which it set to all new objects.

- **"check"** flag is used for turning on/off the automatic design rule checking the created primitives. The file lw.rul or file specified in CHECK menu is used for design rule checking. The checking is done in area outlined by created primitive contour.

- **"neutral"-"auto"-"property"** switcher is used for setting properties for electrical circuit checking. When the switcher have **"neutral"** value, properties are does not created. In **"auto"** mode the editor searches property in other primitives in the place of created primitive and set found property to the created primitive. When you set the **"property"** mode of this switcher, the editor will prompt you to enter the property string for each created primitive.

ARCS menu

This menu variant contains commands for creating arc primitives.

Figure 39 The ARCS Menu

Description of menu switches

Only one menu switcher is contained in the ARCS menu. This is the **"back"** switcher, which provides switching to previous menu variant.

Description of commands

- **"width"** is used for specifying width and end margin for ring and arc. After you pick this item, the editor prompts you to enter these values.

- **"circle"** is used for creating circles.

- **"ring"** is used for creating ring.

- **"arc"** is used for creating arc.

Description of flags

Only one of flags is contained in this menu variant. This is **"8 sides"-"56 sides"** switcher, which is used for specifying the number of segments for arc primitives.

CELL menu

This menu variant is used for placing cells both in normal view and in rotated or mirror reflected view. It is also used for placing text strings into layout.

```
┌─────────────┐
│ CELL        │
│ back        │
├─────────────┤
│ array       │
│ mirror      │
│ Wmirror     │
│ Xmirror     │
│ Ymirror     │
│ 0 s         │
│ scale       │
│ neutral     │
├─────────────┤
│ cellname    │
│ insert      │
│ text        │
│ font        │
│ height      │
│ dbase ls    │
│ director    │
│ OS cmd      │
└─────────────┘
```
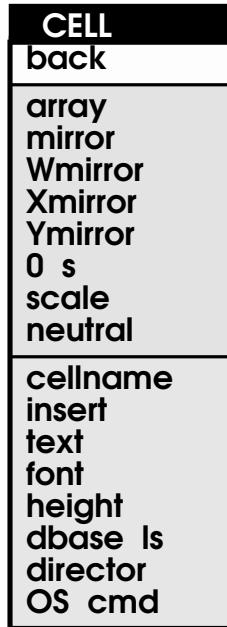
Figure 40 The CELL Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

- **"cellname"** is used for specifying the name of instantiated cell. After you pick this menu command, the editor prompts you to enter the name of instantiated name.

- **"insert"** is used for placing instances into layout. If this command is in active state, you can place current instance by selecting the base point of instance into cell layout.

- **"text"** is used for placing text strings into layout. After you pick this command and select the base point of string placement (left and lower corner of surrounding rectangle), the editor prompts you to select the layer, on which the string will be placed, and then to enter the text string.

- **"font"** is used for specifying the layout font, by which the text strings will be drawn. After you pick this menu

command, the editor prompts you to enter the name of desired font.

- **"height"** is used for specifying the height of created text strings. After you pick this menu command, the editor prompts you to enter the text height you want. By default the height is equal to 100.

  Descriptions of other auxiliary commands see above in the MAIN menu description.

Description of flags

- **"array"** flag is used for multiplying instances. After you set this flag active, the editor prompts you to enter desired array parameters, and then, when you will done the instance placing command, the whole array of instances will be placed instead of single instance.

- **"Wmirror"** flag in active state allows you to place objects (instances or text strings) to side of desired layout. You may place objects both to normal side of layout and to back side (i.e. with orientation reflecting in mirror from normal). To place objects to the back side of layout, you must not only activate this flag, but also switch to mirror reflecting mirror.

  If this flag is passive, objects will be placed to the front side of layout independently of mirror view mode of layout.

- **"Xmirror"**, **"Ymirror"** flags are used for mirror reflecting of layout on both horizontal and vertical directions.

- **"0 s"-"90 s"-"180 s"-"270 s"** flag switcher is used for specifying orientation of placed objects.

- **"scale"** flag is used for placing magnified (or diminished) instance. When you activate this flag, the editor prompts you to enter the zooming coefficient (as proper fraction).

- **"neutral"-"element"-"property"** flag switcher is used for setting mode of adding properties to the placed instances. In **"neutral"** state instances are placed without properties. In **"element"** mode after selecting the base point of instances the editor prompts you to enter the name of element and identifiers of element's ports, starting with 1. in **"property"** state after you select the base point of instance, the editor prompts you to enter the property string.

SELECT menu

This menu variant contains commands, which allows you to select objects for editing in further.
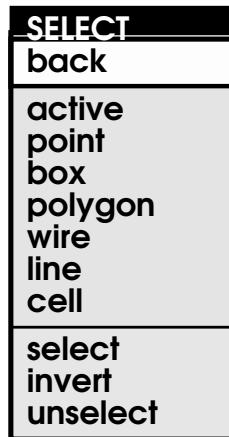

Figure 41 The SELECT Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

- **"select"** command is used for object selecting by specified ways.

- **"invert"** command is used for inverting of object select state. I.e. if the object is already selected and you select it again, it becomes non-selected.

- **"unselect"** command is used for unselecting selected objects.

Description of flags

- **"active"-"visible"-"all"** flag switcher is used for specifying the set of layers, which contains selected objects.

- **"point"-"area"-"global"** flag switcher is used for specifying the mode of object selecting.

- **"box"** flag in active state specifies, that the boxes can be selected.

- **"polygon"** flag in active state specifies, that the polygons can be selected.

- **"wire"** flag in active state specifies, that the wires can be selected.

- **"line"** flag in active state specifies, that the lines can be selected.

- **"cell"** flag in active state specifies, that the cells can be selected.

TRANSFORMATION menu

This menu variant is one of the edit menu variants. Command of this menu allows you to move, copy, delete and deform objects.

```
TRANSFOR
main
file I/O
create
change
sp.trans
select

active
point
same lay
resize
check
ortogona

move
copy
delete
deform
```
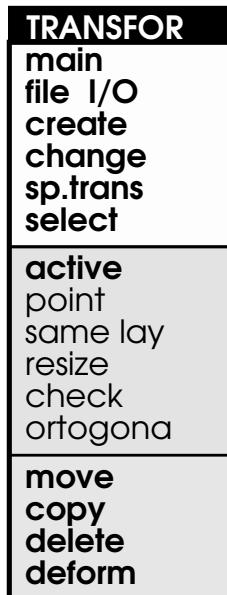
Figure 42 The TRANSFORMATION Menu

Description of menu switches

- **"main"** is the switch to the MAIN menu.

- **"file I/O"** is the switch to the FILE menu.

- **"create"** is the switch to the CREATE menu.

- **"change"** is the switch to the CHANGE menu.

- **"sp.trans"** is the switch to the SPECIAL TRANSFORMATION menu.

- **"select"** is the switch to the SELECT menu.

Description of commands

- **"move"** is used for moving selected objects.

- **"copy"** is used for copying selected objects.

- **"delete"** is used for deleting selected objects.

- **"deform"** is used for deforming (moving selected vertices of objects).

Description of flags

- **"active"-"visible"-"all"** flag switcher is used for specifying the set of layers, which contains selected objects.

- **"point"-"area"-"global"** flag switcher is used for specifying the mode of object selecting.

- **"same layer"-"new layer"** flag switcher is used for specifying layer, on which copied or moved objects will be placed. In **"same layer"** state edited objects will be placed on same layers, on which they are placed before editing. In opposite case edited objects will be placed on another layer (the editor prompts you to select target layer when editing).

- **"resize"** flag in active state specifies, that when moving or copying object sizes will be changed. When you activate this flag, the editor prompts you to enter the value of changing. If this value will be less than zero, object sizes will decrease, if it more that zero, object sizes will increase. By default the value of changing is equal to zero.

- **"check"** flag is used for turning on/off the automatic design rule checking the created primitives. The file lw.rul or file specified in CHECK menu is used for design rule checking. The checking is done in area outlined by created primitive contour.

- **"ortogonal"** flag turns on/off the deforming mode, when the vertices are moved both on vertical or on horizontal directions.

SPECIAL TRANSFORMATION menu

This menu variant contains same commands as in the TRANSFORMATION menu (without **"deform"** command). The difference between these menu variants is that in SPECIAL TARNSFORMATION menu you can also rotate objects, reflect them in mirror, multiply and zoom them.

```
SP.TRANS
back

active
point
same lay
resize
check
scale
array
0 s
Xmirror
Ymirror

move
copy
delete
```

Figure 43 The SPECIAL TRANSFORMATION Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

- **"move"** is used for moving selected objects.

- **"copy"** is used for copying selected objects.

- **"delete"** is used for deleting selected objects.

Description of flags

- **"active"-"visible"-"all"** flag switcher is used for specifying the set of layers, which contains selected objects.

- **"point"-"area"-"global"** flag switcher is used for specifying the mode of object selecting.

- **"same layer"-"new layer"** flag switcher is used for specifying target layer for edited objects (when moving or copying them).

- **"resize"** flag in active state specifies change value of edited objects.

- **"check"** flag is used for turning on/off the automatic design rule checking the created primitives. The file lw.rul or file specified in CHECK menu is used for design rule checking. The checking is done in area outlined by created primitive contour.

- **"scale"** flag in active state specifies zooming of edited objects. After you activate this flag, the editor prompts you to enter the zooming coefficient as proper fraction (first - enumerator and then - denominator with space between them).

- **"0 s"-"90 s"-"180 s"-"270 s"** flag switcher is used for specifying orientation of edited objects.

- **"Xmirror"**, **"Ymirror"** flags are used for mirror reflecting of edited objects on both horizontal and vertical directions.

CHANGE menu

With commands of this menu variant you can execute some specific operations over objects.

```
CHANGE
main
file I/O
create
transfer
select

active
point
45 s
check

uncell
unarray
property
p.divide
delete
```
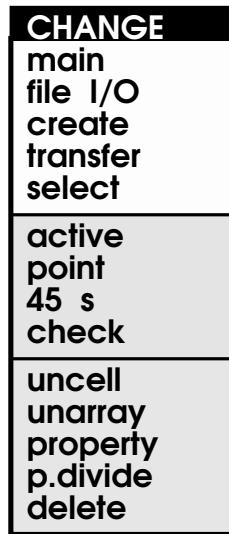
Figure 44 The CHANGE Menu

Description of menu switches

- **"main"** is the switch to the MAIN menu.

- **"file I/O"** is the switch to the FILE menu.

- **"create"** is the switch to the CREATE menu.

- **"transform"** is the switch to the TRANSFORMATION menu.

- **"select"** is the switch to the SELECT menu.

Description of commands

- **"uncell"** is used for flattening cells.

- **"unarray"** is used for changing the array of objects to the set of single objects.

- **"array"** is used for changing the multiply parameters of array.

- **"property"** is used for changing properties of selected objects.

- **"p.divide"** is used for exploding wires or lines in the selected vertices of object path.

- **"delete"** is used for deleting selected objects.

Description of flags

- **"active"-"visible"-"all"** flag switcher is used for specifying the set of layers, which contains selected objects.

- **"point"-"area"-"global"** flag switcher is used for specifying the mode of object selecting.

- **"45 s"-"90 s"-"free angle"** flag switcher is used for correction of angles between neighbouring segments of edited wires, lines and polygons.

- **"check"** flag is used for turning on/off the automatic design rule checking the created primitives. The file lw.rul or file specified in CHECK menu is used for design rule checking. The checking is done in area outlined by created primitive contour.

CHECK menu

The CHECK menu variant is used for design rules checking.

Figure 45 The CHECK Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

- **"rules file"** is used for specifying the file name with rules, which are used in all menu variants. After you pick this command, the editor prompts you to enter the name of DRC file. By default this is file named lw.rul.

- **"err.layer"** is used for specifying the layer, where error primitives (i.e. primitives, which are placed incorrectly in correspondence with design rules) will be outlined.

- **"size"** is used for specifying the value of size for checking the current rule. After you pick this command, the editor prompts you to enter the value of checking size.

- **"del.errs"** is used for deleting error primitives from error layer. After you pick this command, select window, in which you want to delete primitives. Then all primitives will be removed from error layer.

- **"check"** is used for check performing. After you activate this command, select window or area (in dependence of **"global"-"area"** flag switcher states). Then you must select the check layer. You must also select second layer for

another primitive (except **"size"** rule). All primitives are select in the layer window.

After that the editor performs checking.

Description of flags

- **"global"-"area"** flag switcher is used for specifying checked area. In **"global"** state all cell will be checked. In opposite case only area selected by "rubber" rectangle will be checked.

- **"current"-"hierarchy"** flag switcher is used for specifying depth of cell hierarchy for checking. In **"current"** state only current state is checked. In **"hierarchy"** state all cells from hierarchy are checked.

- **"min"-"max"** flag switcher is used for specifying the criterion for checking.

- **"rules"-"size"-"distance"-"el.list"-"overlap"-"inclusion"-"gate ext"-"gate dis"** flag switcher is used for specifying the current rule or file rules, if the **"rules"** state is set.

- **"err.file"** flag in active state specifies, that all information about check errors will be saved in the error file. The name of error file is specified, when you activate this flag and the editor prompts you to enter the error file name.

- **"stop"** flag in active state specifies, that the editor will break after completing each rule and ask you: continue checking or not?

SETUP menu

This menu variant is used for setting and deleting layout windows and changing database parameters.

```
SETUP
back

window
subwindo
delwindo
w-list
init DB
```
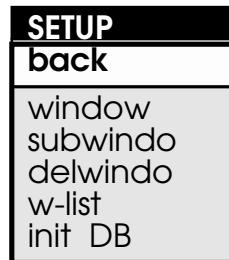
Figure 46 The SETUP Menu

Description of menu switches

The **"back"** command is used for returning to the previous menu variant.

Description of commands

- **"window"** is used for creating new layout window.

- **"subwindow"** is used for creating additional layout window.

- **"delwindow"** is used for deleting existing layout window.

- **"w-list"** displays list of all existing windows (but only layout windows) in dialog window.

- **"init DB"** is used for changing database parameters: the sector size on X direction, the sector size on Y direction and dividing coefficient. After you pick this command, the editor prompts you to enter desired values of these parameters.

  Warning: after you change database parameters, all database contents will be removed.

Section 6
Miscellanea

Creating layout font

To create new layout font, you must set directory, which contains LW, as current by using the **"OS cmd"** command.

Then you must create the directory with name, which will be the name of created font. Files of the created font will be contained in this directory. Switch to created directory by using the **"OS cmd"** command.

You may also copy files of the existing font for creating the new font based on existing font.

Each symbol of font is contained in its single file. The name of symbol file must be the same as the font name, and the file extension must contain the octal ASCII code. For example, the letter **"A"** of format named **"new"** with the ASCII number 101 will be stored in the file new.101. The ASCII numbers of symbols are shown in the table below.

0		20		40		60	0	100	@	120	P	140	`
160	p											
1		21		41	!	61	1	101	A	121	Q	141	a
161	q											
2		22		42	"	62	2	102	B	122	R	142	b
162	r											
3		23		43	#	63	3	103	C	123	S	143	c
163	s											
4		24		44	$	64	4	104	D	124	T	144	d
164	t											
5		25		45	%	65	5	105	E	125	U	145	e
165	u											
6		26		46	&	66	6	106	F	126	V	146	f
166	v											
7		27		47	'	67	7	107	G	127	W	147	g
167	w											
10		30		50	(	70	8	110	H	130	X	150	h
170	x											
11		31		51	)	71	9	111	I	131	Y	151	i
171	y											
12		32		52	*	72	:	112	J	132	Z	152	j
172	z											
13		33		53	+	73	;	113	K	133	[	153	k
173	{											
14		34		54	,	74	<	114	L	134	\	154	l
174													

```
        15          35          55 -      75 =   115 M   135 ]    155 m
175 }
        16          36          56 .      76 >   116 N   136 ^    156 n
176 ~
        17          37          57 /      77 ?   117 O   137 _    157 o
177
```

```
200 À   220 Ð   240 à   260 _   300 _   320 _   340 ð
360 _
201 Á   221 Ñ   241 á   261 _   301 _   321 _   341 ñ
361 ±
202 Â   222 Ò   242 â   262 _   302 _   322 _   342 ò
362
203 Ã   223 Ó   243 ã   263 ¦   303 _   323 _   343 ó
363
204 Ä   224 Ô   244 ä   264 _   304 _   324 _   344 ô
364
205 Å   225 Õ   245 å   265 _   305 _   325 _   345 õ
365
206 Æ   226 Ö   246 æ   266 _   306 _   326 _   346 ö
366
207 Ç   227 ×   247 ç   267 _   307 _   327 _   347 ÷
367
210 È   230 Ø   250 è   270 _   310 _   330 _   350 ø
370
211 É   231 Ù   251 é   271 _   311 _   331 _   351 ù
371
212 Ê   232 Ú   252 ê   272 _   312 _   332 _   352 ú
372
213 Ë   233 Û   253 ë   273 _   313 _   333 _   353 û
373
214 Ì   234 Ü   254 ì   274 _   314 _   334 _   354 ü
374 n
215 Í   235 Ý   255 í   275 _   315 _   335 ¦   355 ý
375 ²
216 Î   236 Þ   256 î   276 _   316 _   336 _   356 þ
376
217 Ï   237 ß   257 ï   277 _   317 _   337 _   357 ÿ
377
```

Customizing LW interface

In LW there are several configure files. They are considered below.

Lower level configure files

Configure files of lower level has the *.cap extension. They contain keyboard and button maps, color map, fill patterns, line styles, current font (not layout font) and cursor pattern. Usually there are two files with *.cap extension:

- file with same name as your operating platform (ms.cap - MS DOS, mswin.cap - Microsoft Windows (NT), x11.cap - X Window System, gpr.cap - Apollo Domain Graphics). Usually this file contains only the

keyboard and button maps, which are machine-related data.

- h.cap file contains common information, which is not machine-related data.

The first described file must also contain the string of including the h.cap file:

    <h.cap

This including place may be in arbitrary part of file.

Menu configure file

The configure menu file has the *.mnu extension. It contains colors and text string for each menu item (for menu switchers these parameters are contained for each switcher state).

Message configure file

The message configure file has the *.mlb extension. It contains all messages using in the editor. You can't change this file interactively, because it is binary file. But, this file is created by converting text file with *.tlb extension, which is the text file. Therefore, you can change file with *.tlb extension (usually lw.tlb) and then convert it (by makemess utility). The string format of *.tlb file is following:

    ##*mess_index*## *mess_string*##

where

- *mess_index* is the internal decimal index of message (you must not change it),

- *mess_string* is the text string of message. The length of string must be less than 500 characters.

At the end of file the following sample string with non-decimal index field may be used:

    ##end##

To convert *.tlb file to *.mlb file, you must enter in command line following string

    > **makemess** *file.mlb file.tlb max_of_records*

where

- *file.mlb*, *file.tlb* are names of target binary file and source text file correspondingly,

- *max_of_records* is the number of records (the index of last record).

Window configure file

This file has the *.win extension. It contains the information about windows, their placement and contents. The format of window description in this file is follows:

"*win_name*"(m=*mode*, bg=*bg_col*, fr=*fr_col*, hl=hl_col) = *geometry* []

where

- *win_name* is the text string corresponded with this window. For menu and layer windows this string is the name of corresponding configure file. For layout windows this string is the last opened cell name.

- *mode* is the internal mode index, which the editor is set.

- *bg_col*, *fr_col*, *hl_col* are the indexes of background, foreground and highlight colors in the window.

- *geometry* is the text string (in X Window System geometry format), which specifies the placement of window in the screen.

The additional information (usually for layout and control windows) may be placed within the square brackets [] at the end of window description.

Appendices

Appendix A
Fill patterns, line styles and colors

Figure 47 shows the detail fill patterns used in LW. They are placed in the sequence order while changing the fill pattern for layer.
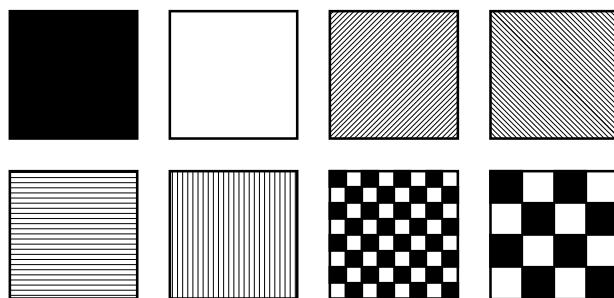
Figure 47 Fill Patterns

Figure shows the detail line styles placed in the sequence order while changing the line style for layer.
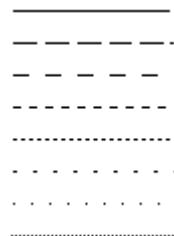
Figure 48 Line Styles

The following listing is the table of base colors used in LW. Colors are placed in the table in the sequence order while changing the current color for layer.

0 – black	8 – dark grey
1 – blue	9 – bright blue
2 – green	10 – bright green
3 – cyan	11 – bright cyan
4 – red	12 – bright red
5 – magenta	13 – bright magenta
6 – brown	14 – bright yellow
7 – white	15 – bright white

Warning: some graphical enviroments can't exactly support resources shown above.

Appendix B
Sample layer table file

The following listing is the sample layer table file.

```
25 layers, 19 in use:
(mode=2,color=11,filling=1,style=0)=00=CALL
(mode=0,color=0,filling=7,style=0)=20=TEXT
(mode=2,color=3,filling=7,style=0)=18=OVER
(mode=2,color=14,filling=4,style=3)=17=MET2
(mode=0,color=15,filling=0,style=0)=16=VIA
(mode=2,color=1,filling=5,style=0)=15=MET1
(mode=3,color=10,filling=1,style=5)=12=PWIN
(mode=2,color=9,filling=0,style=0)=13=NWIN
(mode=2,color=12,filling=0,style=0)=14=CONT
(mode=2,color=6,filling=7,style=0)=11=POLY
(mode=0,color=8,filling=7,style=0)=10=HIPO
(mode=0,color=29,filling=7,style=0)=09=FLDI
(mode=2,color=10,filling=1,style=3)=08=ACTV
(mode=2,color=15,filling=7,style=0)=07=BASE
(mode=2,color=3,filling=0,style=0)=06=NWLL
(mode=2,color=6,filling=7,style=0)=05=PPLG
(mode=2,color=4,filling=7,style=0)=03=NTUB
(mode=2,color=5,filling=7,style=0)=04=NPLG
(mode=0,color=3,filling=7,style=0)=01=BLN
```

Appendix C
Default window configure file

The following listing is the default window configure file named lw.win. This variant of configure file may be used for IBM PC with VGA adapter (640x480 pixels) or for another graphic systems, which have the window size more than 640x480 pixels. To create this file manually at the beginning of edit session, set **/w** switch when editor starting.

```
"Dialog"(m=201,bg=8,fr=7,hl=15)=338x41+298+434
"Cursor"(m=201,bg=6,fr=14,hl=14)=202x41+90+434
"lw.mnu"(m=120a,bg=1,fr=7,hl=15)=82x249+4+5
"lw.ltb"(m=a05,bg=7,fr=2,hl=15)=82x171+4+258
"lw.wvc"(m=2205,bg=14,fr=1,hl=15)=82x41+4+434
"ADC"(m=400,bg=7,fr=2,hl=15)=326x425+90+5
 [l=1237,r=10337,b=4038,t=15910,h=-1,#1,*2,10:
 (100,100)(100,100)(1000,1000)(25,25)
 (50,50)(100,100)(250,250)(1000,1000)
 (2500,2500)(5000,5000)]
"ADC"(m=400,bg=7,fr=2,hl=15)=217x173+419+257
 [l=2078,r=9638,b=7073,t=13093,h=-1,#1,*1,10:
 (1,1)(5,5)(10,10)(25,25)
 (50,50)(100,100)(250,250)(1000,1000)
 (2500,2500)(5000,5000)]
"ADC"(m=400,bg=7,fr=2,hl=15)=217x249+419+5
 [l=-2302,r=13466,b=227,t=18331,h=0,#1,*0,10:
 (1,1)(5,5)(10,10)(25,25)
 (50,50)(100,100)(250,250)(1000,1000)
 (2500,2500)(5000,5000)]
```

Appendix D
Default DRC file

The following listing is the default design rule checking
file that is supplied with installation set of LW. The name
of this file is lw.rul.

```
CIF
;
; 1 unit is 0.1 micrometer
;
3A = NTUB [   60min ]
3B = NTUB )   60min (
3C = NTUB ] 240min [
3D = NTUB )   10max ) BLN
3E = NTUB )   90min ( BLP
;
4E = NPLG ) 320min ( NTUB
;
6A = NWLL [   60min ]
6B = NWLL )   60min (
6C = NWLL ] 240min [
6D = NWLL ) 240min ( NTUB
6E = NWLL ) 320min ( NPLG
6F = NWLL ) 260min ( BLN
6G = NWLL ( 140min ( PPLG
6H = NWLL )    0min ) NTUB
;
17A = MET2 [   50min ]
17B = MET2 )   40min (
17C = MET2 (   20min ( VIA
;
18A1 = OVER ) 100min ) MET1
18A2 = OVER ) 100min ) MET2
;
; End of rules
```

Appendix E
SOURCE format description

The following text is included from document named *"ÏÓËÜÒ. Èåðàðõè÷åñêàÿ ñèñòåìà ðåäàêòèðîâàíèÿ òîïîëîãèè ÁÈÑ. Âåðñèÿ 1.3. Ðóêîâîäñòâî ïîëüçîâàòåëÿ."* KIPARIS Group doesn't carry the responsibility for non-correct description of format and layout, imported from other systems.

The file of SOURCE format is binary file quoted by 512 bytes. The coding of layout is performed on base of basic elements of format. Each record starts with service word (passport). It contains the length (in words) in first byte and the code of element in the second byte.

Following passport additional information is usually placed. Numerical data is coded in DEC decimal (integer) style.

Box with sides parallel to co-ordinate axes

3	9
layer	
Y1	
X1	
Y2	
X2	

3 is the code of this object

9 is the length (in words) of object description

layer is the layer number

Y1, X1, Y2, X2 are the box co-ordinates

Box with arbitrary slope of sides

9	17
layer	
Y1	
X1	
Y2	
X2	
Y3	
X3	
Y4	
X4	

9 is the code of this object

17 is the length (in words) of object description

layer is the layer number

Y1, X1, Y2, X2, Y3, X3, Y4, X4 are the box co-ordinates

Polygon

X	23	L
layer		
unused		
AminY		
AminX		
AmaxY		
AmaxX		
Y1 (<)		
X1 (<)		
· · · · · · · · · ·		
Yk (<)		
Xk (<)		

X is direction of description:

- 1 means the clockwise description

- 0 means the counter-clockwice description

23 is the code of polygon

L is the length (in words) of object description

layer is the layer number

AminY, AminX, AmaxY, AmaxX are offsets in bytes relatively to record beginning to elements, which are contained minimum and maximum values of polygon co-ordinates

Y1, X1 are co-ordinates of first vertex

Yk, Xk are co-ordinates of last vertex

All co-ordinates are shifted to the left on 3 digit and junior digits will be set by following way:

- **0** means that both X and Y are changed

- **1** means that only Y is changed

- **2** means that only X is changed

Warning: the **Yk** must be equal to **Y1** and **Xk** must be equal to **X1**.

Line

Format of lines is the same as the polygon format with following exceptions:

1. AminY = 1085; **AminX, AmaxY, AmaxX** are not used.

2. First and last vertices may be not coincide

Wire or polygon with non-changed width

X	7	L
layer		
Y1		
X1		
. . . .		
Yk		
Xk		

- **1** means the clockwise description

- **0** means the back clockwise description

7 is the code of polygon

L is the length (in words) of object description

layer is the layer number

Y1, Y1 are co-ordinates of first vertex of track

Yk, Xk are co-ordinates of last vertex of track

The segment **(X1,Y1)-(X2,Y2)** specifies the width of track (wire width)

Warning: the first and last vertices must be coincide.

X is direction of description:

Valid element

0	L
. . . .	
. . . .	
.	
. . . .	
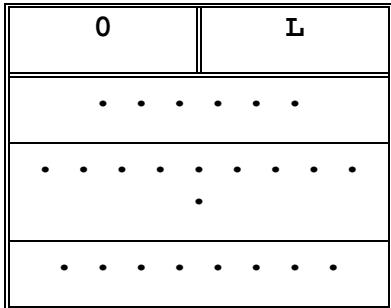
This element is used on the boundary of blocks

0 is the code of valid element

L is the number of valid words

Library element

2	8
name	
R	
Y	
X	
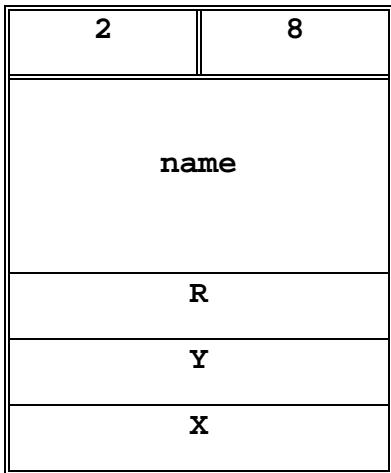
2 is the code of library element

8 is the length (in words) of object description

name is the element name (4 characters)

Y, **X** are co-ordinates of base point of element

R specifies the orientation of element

Array of library elements

10	12
name	
R	
Y	
X	
KY	
LY	
KX	
LX	

10 is the code of array of library element

12 is the length (in words) of object description

name is the element name (4 characters)

Y, **X** are co-ordinates of base point of element

R specifies the orientation of element

KY, **KX** specify the number of elements on co-ordinate axes

LY, **LX** specify the number of elements on co-ordinate axes

Text

1	L
layer	
Y	
X	
R	
S2	S1
. . . .	
Sk	S(k-1)
H	
T	

1 is the code of text

L is the length (in words) of object description

layer is the layer number

Y, **X** are co-ordinates of base point of element

R specifies the orientation of element

Si are symbols of string in ASCII codes. If one of these bytes will be equal to zero, the end of string is achieved.

H specifies the text height

T is flag, which specifies the font of text:

- **1** means, that there is vector font is used

- **2** means, that there is contour font is used

Service block

4	L
. . . .	
. . . .	
.	
. . . .	

4 is the code of service block

L is the number of valid words

This element is used as the end block of file. It does not filled.

Orientation codes

- **0** is source orientation

- **1** means rotating to the left on 90 s.

- **2** means rotating to the left on 180 s.

- **3** means rotating to the right on 90 s.

- **9** means mirror reflecting relatively X axes

- **10** means mirror reflecting relatively X axes and rotating to the left on 90 s

- **11** means mirror reflecting relatively X axes and rotating to the left on 180 s

- **12** means mirror reflecting relatively X axes and rotating to the right on 90 s

Appendix F
CIF format description

This text is included from the book of *Mead C., Conway L. Introduction to VLSI Systems. ADDISOW-WESLEY, 1980.* More formal description of format is contained in the book of *B. Hon, C.H.Sequin Guide to LSI Implementation (2nd revised and extended edition). XEROX PARC, Palo Alto, CA, Jan. 1980.* KIPARIS Group doesn't carry the responsibility for non-correct description of format and layout, imported from other systems.

The file with layout description consist of ASCII symbol set. The file contains the set of commands, which are finished by semicolon.

The following list is the formal description of CIF 2.0 format in the Virt's text notation. There are special notation symbols:

- **|** is OR

- **"** is separator of terminal symbols

- **{}** provides the construction repeating zero or more times

- **[]** specify condition factors

- **()** group the construction

- **;** is the end of rule

The formal description of CIF 2.0 format is:

```
FILE = { {BLANK} X_COMM SEMI }
        {BLANK} N_COMM SEMI
        {BLANK} DEF_START_COMMAND SEMI
        { {BLANK} [COMMAND] SEMI }
        {BLANK} DEF_FINISH_COMMAND SEMI
        {BLANK} END_COMMAND {BLANK).
```

```
DEF_START_COMMAND="D" {BLANK} "S" INT [ SEP INT SEP
INT ].
```

```
DEF_FINISH_COMMAND = "D" {BLANK} "F".
```

```
COMMAND = P_COMM | B_COMM | W_COMM | L_COMM |
U_COMM |              X_COMM | CALL_CONSTR
```

```
P_COMM = "P" PATH.
```

```
B_COMM = "B" INT SEP INT POINT.

W_COMM = "W" INT SEP PATH.

L_COMM = "L" {BLANK} SHORT_NAME.

U_COMM = DIGIT USER_TEXT.

X_COMM = "(" COMMENT_TEXT ")".

N_COMM = DIGIT FILE_SPEC.

C_COMM = "C" INT [{BLANK} "M" {BLANK} "X"]
                [{BLANK} "M" {BLANK} "Y"]
                [{BLANK} "R" POINT]
                {BLANK}  "T" POINT.

CALL_CONSTR = N_COMM SEMI {BLANK} C_COMM.

PATH = POINT { SEP POINT }.

POINT = SIGN_INT SEP SIGN_INT.

SIGN_INT = {SEP} [ "-" ] INT_D.

INT = {SEP} INT_D.

INT_D = DIGIT {DIGIT}.

SHORT_NAME = CHAR [CHAR] [CHAR] [CHAR].

CHAR = DIGIT | UPPER_CHAR.

UPPER_CHAR = "A" | "B" | ... | "Z".

DIGIT = "0" | "1" | "2" | "3" | "4" | "5" | "6" |
"7" | "8"        | "9".

SEMI = {BLANK} ";" {BLANK}.

SEP = UPPER_CHAR | BLANK.

USER_TEXT = {USER_CHAR}.

FILE_SPEC = {CHAR} ["." {CHAR}].

COMMENT_TEXT = {COMMENT_CHAR} | COMMENT_TEXT "("
                      COMMENT_TEXT ")"
COMMENT_TEXT.

BLANK = " ". (space)
```

USER_CHAR = Any non-system ASCII symbol, except
";".

COMMENT_CHAR = Any non-system ASCII symbol, except
"(",   ")".

The sense contents of constructions is describe below:

P_COMM is polygon description. PATH is contour description.
The coincidence of first and last points is optional.

B_COMM is description of box with sides parallel to co-
ordinate axes. The first two numbers are the width and
height of box, and last two numbers are co-ordinates of box
centre.

W_COMM is description of wire with non-changed width. The
first number is the wire width. PATH is set of path
vertices.

L_COMM is description of layout layer. All the following
primitives are belong to this layer.

U_COMM is description of the user-defined command.

X_COMM is description of comments. Comments can't be placed
inside commands.

N_COMM is description of the file specification. It is not
recommended to include device and directory path into
specification..

C_COMM is description of instance. The order of parameters
determines the order of co-ordinate transformations. MX, MY
is mirror reflecting on X and Y directions correspondingly.
R with following (x, y) vector determines angle of
clockwise rotation, where x means cosine and y means sine.
T is the translation point.