7 Place and Route File Formats

CAP Files

L-Edit/ SPR calculates the nodal capacitances and other characteristics of the interconnect and outputs this information to a nodal capacitance file with the extension **.cap**.

Syntax

The following is an example of a CAP file.

```
$ ------------------------------------------------------------------------------
$ Nodal Capacitance File       :  D:\ledit_files\v8_shipping\example3\bargraph.cap
$ SPR Date and Time            : 02/27/1999 - 8:00
$
$ H1 layer-to-substrate cap. - Area : 36 aF/sq.micron   Fringe : 0.086 fF/micron
$ V  layer-to-substrate cap. - Area : 11 aF/sq.micron   Fringe : 0.077 fF/micron
$ H2 layer-to-substrate cap. - Area : 7 aF/sq.micron   Fringe : 0.031 fF/micron
$
$ H1 layer-to-V  layer cap.  - Area : 31 aF/sq.micron
$ V  layer-to-H2 layer cap.  - Area : 28 aF/sq.micron
$ H1 layer-to-H2 layer cap.  - Area : 10 aF/sq.micron
$
$ 1 Locator Unit (LU) = 1/1 Lambda = 7/20 Micron(s)
$
$
$ Node    Capacitance    No of        Length     Area       Area       Area
$                        Terminals    of Node    on H1      on V       on H2
$         (1/100 pF)                  (LU)       (LU^2)     (LU^2)     (LU^2)
```

```
$ ------------------------------------------------------------------------------

N4        6              9        5964.400     678.0000     10567.200    13296.000
N27       7              2        6046.000     975.0000     12480.000    9366.000
N26       2              2        2452.000     0.0000       4479.000     5754.000
N25       9              2        7589.000     2373.0000    11829.000    17130.000
N24       5              2        4062.000     0.0000       3936.000     16500.000
 :        :              :        :            :            :            :
N12       1              2        599.500      0.0000       1798.500     0.0000

$ ------------------------------------------------------------------------------
$
$ Length of all nets (LU) :       1893584.90
```

Interpretation

Each line in the file is in the format:

node capacitance NoOfTerminals Length AreaOnH1 AreaOnV AreaOnH2

where:

node Name of the node.

capacitance An integer denoting capacitance of this node in hundredth of a picofarad.

NoOfTerminals Number of pins attached to the node.

Length	Length of the interconnect of the node.
AreaOnH1	Area of the route taken by the node on the H1 layer.
AreaOnV	Area of the route taken by this node on the H2 layer.
AreaOnH2	Area of the route taken by this node on the H2 layer.

For a detailed description of how nodal capacitances are calculated, see Output Options on page 2-107.

EDIF Files

Netlist files in Electronic Design Interchange Format (EDIF) are used by SPR and BPR to place and route a design. EDIF netlist files typically have a filename extension of **.edf**, **.edn**, or **.edi**.

L-Edit supports EDIF version 2 0 0 with EDIF level 0, keyword level 0, and netlist view—**(edifLevel 0)**, **(keywordLevel 0)**, and **(viewType NETLIST)**. Other view types are ignored.

The netlist parser is limited to one netlist view label per netlist. If the netlist contains more than one netlist view, L-Edit warns you and ignores subsequent, different view labels.

All cell properties will be transferred to the relating instance. Properties with no relation to a netlist view will be ignored. L-Edit currently supports the use of properties with regard to the labeling and positioning of pads and I/O signals. If an EDIF netlist contains both pad properties and I/O signals, only the pads will be considered. L-Edit provides an optional warning when both appear in a netlist, which may be disabled using **Setup > Application > Warnings**.

The parser is limited to one design—**(design *designname* (...))**–per EDIF netlist file.

External EDIF library definitions—**(external *libraryname* (...))**—are treated in the same way as normal EDIF library definitions—**(library *libraryname* (...))**. Furthermore, the parser considers only library information present in the current EDIF netlist file.

A mapping table is generated if cell names or port names in the EDIF netlist differ from those used in the cell library. L-Edit automatically accesses the **Mapping Table** dialog whenever it encounters a cell or port name discrepancy between the EDIF netlist and the standard cell library. You can also access this dialog via the **SPR Setup** dialog or the **BPR Initialization** dialog. The mapping information is stored in the TDB design file.

Syntax

The following example shows excerpts from an EDIF netlist containing pad cells, I/O signals and critical nets. If pad cells *and* I/O signals are both included in the netlist, the pad cell configuration has precedence over I/O signals.

```
(edif bargraph
(edifVersion 2 0 0)
        (edifLevel 0)
        (keywordMap (keywordLevel 0))
        (status
            (written
                (timestamp 1998 11 01 07 5 00)
                (program "S-Edit" (version "Version 2.06"))))
        (library bargraph_top
            (edifLevel 0)
```

```
(technology (numberDefinition (scale 1 (E 1 -12) (unit CAPACITANCE))))
(cell bargraph_top
    (cellType GENERIC)
    (status
        :
    (view view_1
        (viewType NETLIST)
        (interface
            (port C1B (comment "I/O Signal")
                (property PIN_LOCATION (string "L2"))
                (direction INPUT))
            :
            (instance PadInC_1
                (viewRef view_1 (cellRef PadInC))
                (portInstance DataIn)
                (portInstance DataInB)
                  :
                (property PAD (string "L1")) (comment "Pad")
            :
            (net N54
                (joined
                    (portRef DataInB (instanceRef PadInC_5))
                    :
                )
                (criticality 100) (comment "Net criticality")
            )
            (net N55
                (joined
                    (portRef C1B) (comment "Reference to an I/O Signal")
                    :
                )
            )
```

```
                    :
                )
            ))
        )
        (design ROOT
            (cellRef bargraph_top
                (libraryRef bargraph_top)))
)
```

Interpretation: Pads

Pad cells are defined by creating a property named **PAD** with a value such as **L1** in the pad cell instance. The following formats are supported:

```
(property PAD (string "L1"))
```

or

```
(property PAD (string "1"))
```

The string value determines the position of the pad, counting counterclockwise. (In this example, the pad is placed on the upper-left position of the padframe.) In the first format, the sides of the padframe are labeled with **L** (left), **B** (bottom), **R** (right), and **T** (top). The subsequent number determines the position on this side. The second format labels the pad position with only a numeric value (> 0). The resulting position is determined according to this value, counting counterclockwise, starting from the upper left position on the padframe. In other

words, pad position **L1** is equal to pad position **1**. To avoid ambiguity, it is recommended that only one format be used in the same netlist file.

Pads with no string value attached, e.g., **(property PAD (string ""))**, are equally distributed around the padframe.

Interpretation: I/O Signals

If you intend to generate an SPR core cell only, you don't need to specify pad cells. In this case, ports assigned to the top-level cell in your EDIF netlist can be assigned to I/O signals leaving the core. Please note that I/O signals will only be considered if the netlist does not contain pad cells. (If *both* are present, pad cells will be considered and I/O signals will be ignored.)

I/O signals are defined through the EDIF **interface** keyword. When SPR reads the netlist, it uses the I/O signals designated in the **interface** section to initialize the core I/O signals setup (**SPR > Setup > Core Setup—I/O Signals** dialog). You can change the signal location in this dialog to suit your design.

In most schematic editors, when you define pins or ports to a schematic or symbol, they are designated as I/O signals. The following is an example showing the I/O signal **CLK**:

```
(interface
(port CLK
(direction INPUT)
)
```

In the above example, no pin position has been provided. In this case, the signal **CLK** will be equally distributed with the other I/O signals around the core during placement. (A specific I/O signal position can be defined in the (**SPR Core Setup—I/O Signals** dialog.)

Alternatively, you can also provide an I/O signal port position as shown below:

```
(interface
(port CLK
(direction INPUT)
(property PIN_LOCATION (string "L2"))
)
```

This example will place the I/O signal **CLK** on the second position (top to bottom) on the left side of the core. The EDIF property **PIN_LOCATION** with the property value **1, 2, …** or **L1, L2, ..., B1, … , R1, …, T1…** indicates the relative position in which the I/O signals will be placed around the core, counting counter-clockwise.

Interpretation: Criticality

Critical nets are defined by using the **criticality** construct in EDIF. SPR considers critical nets during placement optimization. Criticality is expressed as an integer value. It may be positive (this net is given a higher priority for placement purposes) or negative (this net is given a lower priority during placement). The default value for any net that is not specified with a criticality value is zero.

The consideration of net criticality in SPR is based on two assumptions:

- The numerical value of the criticality describes the relative importance of a net compared to others. For example, if the criticality of net A is twice the criticality assigned to net B, then the placer considers it twice as important to reduce the length of net A compared to net B.
- The critical values are scaled internally according to the largest value that has been entered, with the largest value assigned to a fixed internal value. For example, if net A is the only net with an assigned criticality, then any criticality value greater than zero for this net would lead to the same result.

Additional Notes

An EDIF netlist file must conform to the following rules:

- An EDIF netlist component (e.g., cell, port) must be defined completely before it can be used.
- If your cell interface in the EDIF file contains ports which are not connected in your design, you can label them as “not used” during the mapping process.
- All signals that are to be routed within the core or from the core to the padframe must be listed, with the exception of VDD and GND signal connections to pads.

- Power and ground pads do not have to be included in the netlist. If they are not included, L-Edit places them automatically in accordance with the power and ground rails.
- The range of integer numbers is $-2^{31}+1 \le x \le 2^{31}-1$ (32-bit signed integers). Real numbers are valid in a range of $-1 \times 10^{35} \le y \le 1 \times 10^{35}$. The length of a string is limited to 256 characters. The length of a line is limited to 512 characters.
- The array construct **(array *arrayname* (...))** in a name definition is limited to one- or two-dimensional arrays.
- Valid EDIF identifiers consist of alphanumeric or underscore characters, and must be preceded by an ampersand (&) if the first character is not alphabetic. Thus, "pure" integer numbers are not allowed as identifiers.
- An ampersand (&) at the beginning of an identifier will be ignored. The case of a character is not significant. For example, **&Nand2**, **Nand2**, and **nand2** all represent the same EDIF name.

References

A complete description of the EDIF standard is contained in the Electronic Industries Association (EIA) publication, Electronic Design Interchange Format Version 2 0 0 (ANSI/EIA Standard 548-1988), Electronic Industries Association, 1988, ISBN 0-7908-0000-4.

SDF Files

Pin-to-Pin Delay Syntax

```
(DELAYFILE
(SDFVERSION "OVI Standard 3.0")
(DESIGN "bargraph")
(DATE "02/22/1999")
(VENDOR "Tanner Research, Inc.")
(PROGRAM "L-Edit/SPR")
(VERSION "8.0")
(DIVIDER /)
(VOLTAGE)
(PROCESS)
(TEMPERATURE)
(TIMESCALE 1ps)
(CELL
       (CELLTYPE "bargraph")
       (INSTANCE bargraph)
       (DELAY
              (ABSOLUTE
                     (INTERCONNECT NAND_1/C NOR_2/A (0.005))
                     (INTERCONNECT NAND_1/C NOR_2/B (0.003))
                            etc.
              )
       )
)
```

```
)
```

Net Delay Syntax

```
(DELAYFILE
    (SDFVERSION "OVI Standard 2.1")
    (DESIGN "bargraph")
    (DATE "10/30/1998-07:50")
    (VENDOR " ")
    (PROGRAM "L-Edit/SPR")
    (VERSION "8.0")
    (DIVIDER.)
    (VOLTAGE)
    (PROCESS)
    (TEMPERATURE)
    (TIMESCALE 1ps)
    (CELL (CELLTYPE "bargraph")
        (INSTANCE )
        (DELAY
            (ABSOLUTE
                (NETDELAY BARGRAPH_1/BG64_2/Buf2_1/N1 (0.0031))
                (NETDELAY BARGRAPH_1/BG64_2/N9 (0.0156))
                    :       :
                (NETDELAY N1 (26.2331))
                (NETDELAY N3 (29.2829))
            )
        )
    )
)
```

Interpretation

The pin-to-pin delay is generated in the OVI SDF Specification Standard 3.0:

INTERCONNECT *<port_instance_1> <port_instance_2> (<delay>)*

port_instance_1 is an output or bi-directional port. ***port_instance_2*** is an input or bi-directional port. ***delay*** is the interconnect delay between the output and the input ports.

The net delay is generated in the OVI SDF Specification Standard 2.1:

NETDELAY *<netname> (<delay_of_this_net>)*

The ***delay_of_this_net*** is the delay of the interconnection containing the longest path from an output or bi-directional port to an input or bi-directional port.

The ***DESIGN*** entry in the SDF file header indicates the name of the design—that is, the name of the TDB file.

The ***CELLTYPE*** entry indicates the name of the cell—either the chip cell name (if pad route is included) or the core cell name (if core route only).

TPR Files

L-Edit can use netlist files in Tanner Place and Route (**.tpr**) format to generate chip layouts. TPR files are ASCII text files that are generated automatically by the schematic editor S-Edit; they can also be created with any text editor.

Syntax

A portion of the **.tpr** netlist file for the bargraph example is shown below.

Comment line

```
$ TPR written by the Tanner Research schematic editor, S-Edit
$ Version: 2.0 Beta 5     Jan 7, 1998  16:07:16
```

Pad cell definition
Instance definition

```
CP PadOut DataOut Pad;
UPadOut_1 N2 PAD_B1_L31;

:

CP PadInC DataIn DataInB DataInUnBuf Pad;
UPadInC_1 N68 IPAD_9/N2 IPAD_9/N1 PAD_L9_SCO;
```

In the two lines above, **DataIn**, **DataInB**, and **DataInUnBuf** are the names of ports in the pad cell **PadInC** (PortList). **N68**, **IPAD_9/N2**, and **IPAD_9/N1** are the names of nets attached to these ports (NetList). **PAD_L9_SCO** is the name given

to the body region of the pad. **L9** identifies the position of the pad as the ninth pad from the top on the left side of the padframe.

```
:
```

Ground pad

```
CP PadGnd Pad;
UPadGnd_1 PAD_R8_GND;
```

Power pad

```
CP PadVdd Pad;
UPadVdd_1 PAD_L6_VDD;
```

```
:
```

Cell definition
Instance definition

```
C INV A Out;
UINV_3 BARGRAPH_1/BG64_2/N9 BARGRAPH_1/BG64_2/SFT3;
```

```
:
```

```
C Mux2 A B Out Sel;
UMux2_1 BARGRAPH_1/BG64_1/BG4_1/N118 BARGRAPH_1/BG64_1/BG4_1/N108 N62
+  BARGRAPH_1/BG64_1/S11;
```

In the three lines above, **A**, **B**, **Out**, and **Sel** are ports in the standard cell **Mux2** (PortList). **BARGRAPH_1/BG64_1/BG4_1/N118**, **BARGRAPH_1/BG64_1/BG4_1/N108**, **N62**, and **BARGRAPH_1/BG64_1/S11** are the names of nets attached to these ports (NetList). Note that these net names include the hierarchical structure of the schematic. This is the manner in which S-Edit creates a “flattened” **.tpr** netlist. A plus sign (**+**) indicates a continuation of the previous line.

Interpretation

Pad cells are defined in the format:

```
CP <padname> <pin1> <pin2> … Pad
U<gateUID> <net1> <net2> … Pad_<PadPosition>
```

Standard cells are defined in the format:

```
C <cellname> <pin1> <pin2> …
U<gateUID> <net1> <net2> …
```

A **.tpr** file must conform to the following rules:

- All signals which are to be routed within the core or from the core to the padframe are required to be listed, with the exception of the Vdd and Gnd signal connections to pads.
- For each cell, the **PortList** and **NetList** must have the same number of elements.
- The name “PAD” in the PortList of a pad cell refers to the actual bonding region of the pad, and is not actually involved in the placement and routing process. Pad cells must have a signal marked “PAD.”
- The bonding region of a pad can contain the location of the pad on the padframe. For example, “B1” stands for the leftmost pad on the bottom side of the padframe (L = Left, B = Bottom, R = Right, T = Top).

- Power and ground pads do not have to be included in the netlist. If they are not included, SPR will place them automatically.
- The parts listed in the file must match the cells contained in the layout library. To match, the name of the part must be identical to the name of the library cell (except for case), and every signal listed in the part description must have at least one port of the same name somewhere in the library cell.