

# Interactive Functions of the Dlegate Software Package

(Translation of Annex A of Martin Schobert's Diploma Thesis)

**Version:**

March 25, 2012

**Translator:**

Roberto Avanzi





# Appendix A

## Interactive Features

The interactivity of the Degate software is one of its essential features. This chapter describes the functionality of the Degate GUI.

### A.1 Configuration of the Layers, Project Settings

The reverse engineering of chips using Degate begins with creating a Project Folder. A Project is a container for image data and metadata relating to a chip or part thereof. Multiple layers in integrated circuits are represented by the fact that projects consists of stack of layers. Each of these layers may contain their own artwork and elements of the “logic model.” Here, a logic model is understood as the totality of all switching elements. In a broader sense non-functional elements are included as well, that only purvey geometric information (e.g. annotation and sub-projects), or serve to represent groups (Modules).

The dialog box *Layer Configuration* (see Figure A.1) manages the configuration of the layers – i.e. their relative placement. It is enabled when the user creates new projects and can be opened using the menu ► Layer → Layer configuration. On this dialog one can select the images for individual layers. Furthermore, the user can decide the type of a layer. Descriptive comment can be attached to layers. It is also possible to *disable* levels: This means that these levels do exist in the logic model, but they are hidden to the user when using the software. This function is useful when it is desirable to manage alternative graphic material for layers already imported into the software. The order of layers can be changed by drag-and-drop.

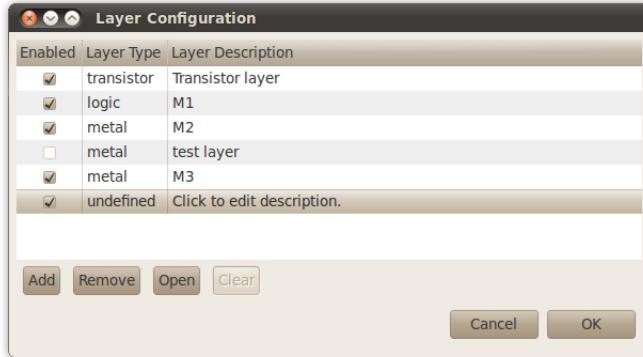


Figure A.1: Configuration of the Layers

When importing multi-layered images, the layers must have the same “coverage.”<sup>1</sup>

Under the menu item ► Project → Project Settings general information about a project can be given (see Figure A.2). A project should have a name and description. In the dialog box

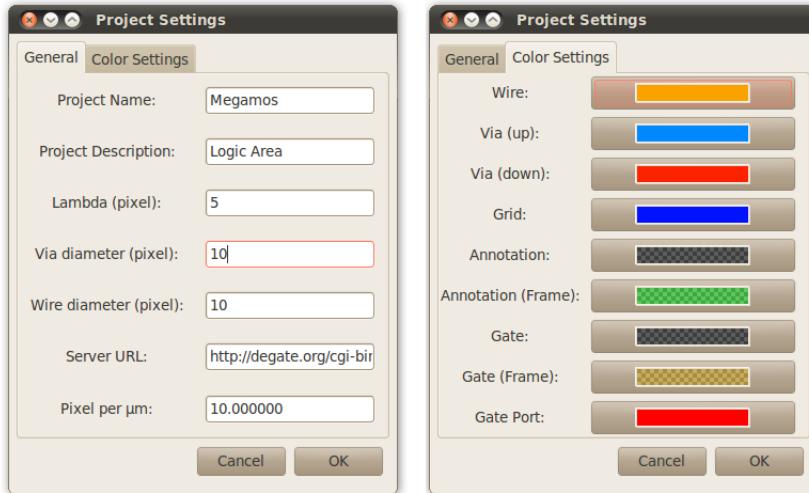


Figure A.2: Project Settings

<sup>1</sup>In a pre-release version of Degate it was possible to define four reference points per layer. Two of the reference points were used to find the corresponding points in the underlying layer. The other two reference points were used to find the corresponding points in the overlying layer. When the four reference points were set for all levels (for the uppermost and lowermost layer, of course, only two), scaling and translation factors were determined automatically for all levels. By means of additional markers it would be then possible to correct for rotational transformations. The function for the alignment of levels, however, was not used, therefore it was not ported to the final version of the software. Most likely simple offsets would be sufficient to determine the relative orientation of the layers.

---

the standard diameter for vias and traces can be set. The configurable parameter  $\lambda$  in this screen is a tolerance value: It indicates by how many pixels two electrically conductive objects are allowed to stay away from each other in order to be still considered contiguous.

The Server-URI is the address of an XML-RPC server that is used in collaborative work to synchronize data. This address is generated automatically upon project creation.

The entry “pixels per  $\mu\text{m}$ ” is used to convert lengths.

Under “Color Settings” the user can choose per-project colors, for instance for text annotations and the representation of traces.

## A.2 Main window, menu and navigation

In the main window (see Figure A.3), the image data of a project is displayed. It will be always shown a single layer. Via buttons on the toolbar and keyboard shortcuts the user can switch to the next or previous layer. Inactive layers are skipped. it is also possible to directly select any level

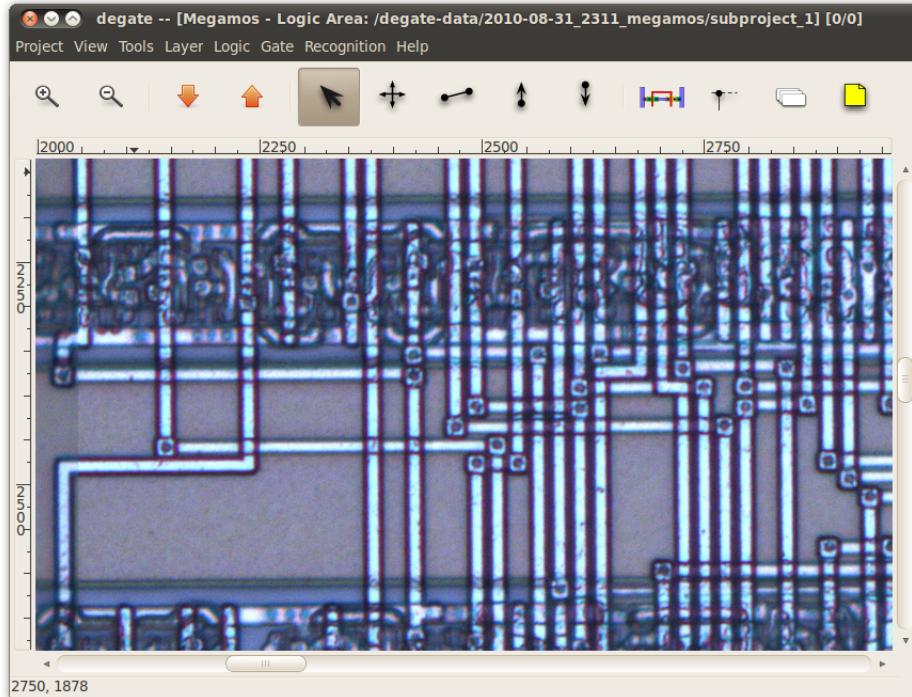


Figure A.3: Main Window

using a keyboard command. Besides keyboard navigation, it is possible to use the mouse and GUI elements in a way entirely similar to most graphics editing programs.

Actions can be invoked via menus. The menu area is divided in the following main entries:

- Under *Project* general project specific are grouped together: the creation of projects and project archives, switching to sub-projects and back, project settings and modifying the configuration settings for sending and receiving data sets to and from the collaboration server.
- The entry *View* collects actions for navigation, and showing/hiding grids and other GUI elements.
- *Tools* allows the selection of tools, which are also available on the Toolbar. These are used to mark areas, select objects, scroll the view, drawing of traces and vias.
- Under *Layer* functions for importing and exporting image data, deleting data and configuration of the image layers are found.
- The *Logic* menu collects actions that deal with the logic model. The include, for instance, actions for electrically connecting and disconnecting or deleting objects, manage annotations of objects and modules, introspection of electrically linked elements and to run rule checks.
- Special logic model functions, which deal primarily with standard cells, are arranged in the *Gate* menu. These include functions for creating standard cells, to place, orient, merge image data; naming, finding, removing, and listing standard cells and choose the color representation of ports.
- Via the *Recognition* menu functions can be invoked for the automatic recognition of standard cell placement, traces and vias.

### A.3 Grid Line Configuration

Auxiliary Grids are project-wide – that is they do not apply to a single layer – and mainly serve to determine the alignment of placed standard cells. They are shown in the main application window.

An auxiliary grid can be either be equally or irregularly spaced and consist of either vertical or horizontal lines – called guides. An equally spaced grid is defined by means of an offset and a spacing value. The spacing value is a floating point number to allow for subpixel accuracy.

---

Irregularly spaced grid lines can be set individually by using the right mouse button in the main window. From a pop-up menu the user selects whether at the clicked position, a horizontal or vertical guide should appear. Figure A.4 shows how they are represented graphically.

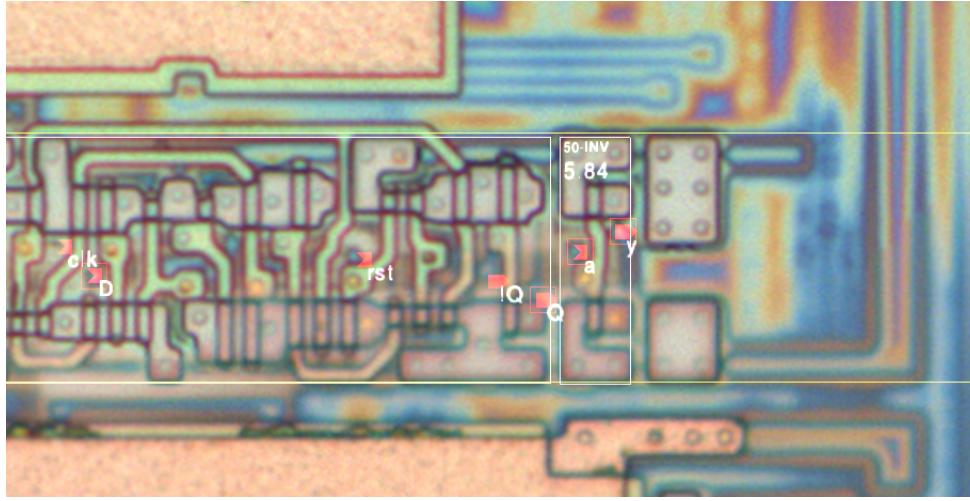


Figure A.4: Representation of Grid Lines along Standard Cells

The actions for configuring the auxiliary grid are found in the menu ► View → Edit Grid. Through this menu all the offsets that define an irregular grid can also be given manually, however this is a less practical way to edit it – see Figure A.5.

Standard cell templates and guides should be aligned, because then it suffices to look for other instances along the same guides. This can considerably restrict the search space. This optimization is discussed in Section 7. In addition, one can use the grid to check whether the pictures contain rotations or other deformations. For instance, waviness can occur as a result of stitching several pictures. The search for standard cell instances is to a certain degree insensitive to waves.

## A.4 Determining and Describing Standard Cells

In Section 3.4 it was described how standard cell layouts can be identified. If a standard cell has been detected, the region that contains it can be marked with the mouse in the main window. ► Gate → Create Gate By Selection. is then used to establish that the selected area corresponds to a standard cell. This opens a dialog window. In Figure A.6 we see an example of a standard 3-cell type NAND. The first entry in the dialog box specifies a short name, which will be

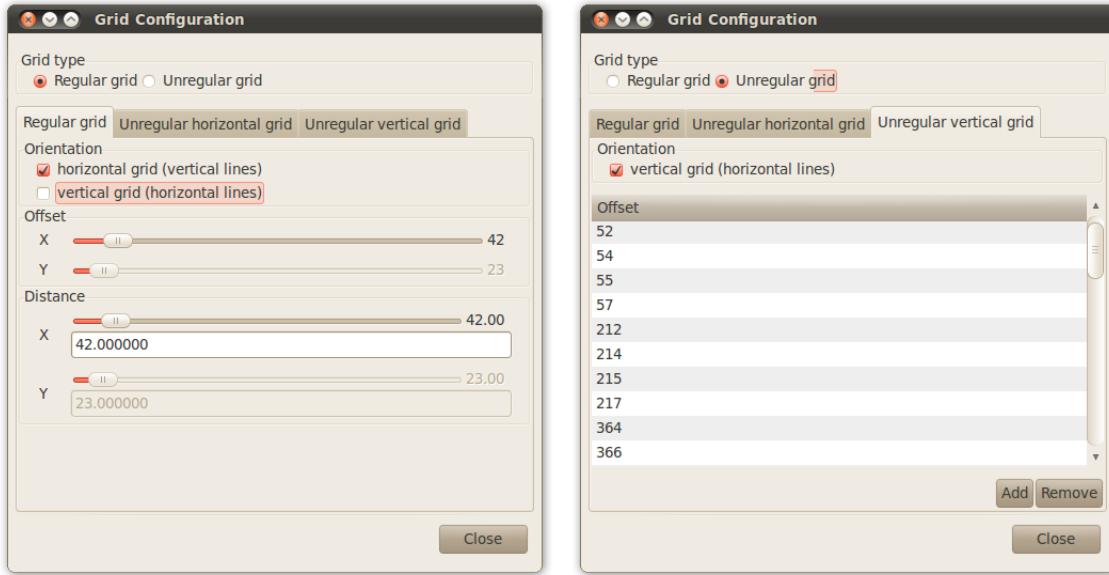


Figure A.5: Configuration of Grid Lines

be displayed in the main window inside the framing of a standard cell. A short name should not contain more than five characters. When this text is shown in the standard cell, it is scaled so that it fits inside the frame. This also means that the text may be too small for reading in the case of a narrow cell. For a detailed explanation the description field can be used.

By selecting “Logic Class”, the standard cell type can be specified. Possible assignments correspond to different logic functions, e.g. (tristate) inverter, (N)AND, (N)OR, X(N)OR. Furthermore, buffers, various types of latches and flip-flops, half and full adders, (de) multiplexers, insulation gates and multistage combinatorial types (AOI) can be specified. This type information is used when generating code frameworks to ensure that an appropriate Template Category is selected.

Each standard cell has one or more ports, i.e. inputs and outputs<sup>2</sup>. These interfaces can be defined under “Ports.” Multivalued logics according to IEEE 1164, including tri-states, can be mapped into Degate. This is done not in the *Description* tab of the entity, but is part of the description of the *Behavior* (cf. Figure A.7).

Colors can be assigned to all objects that can be defined in Degate, including standard cells. The

<sup>2</sup>An isolation gate will be treated in Degate as a special case of a standard cell. At the Register transfer layer, which is relevant for the reconstruction of switching functions, isolation gates are irrelevant. Therefore, they can be there essentially ignored.

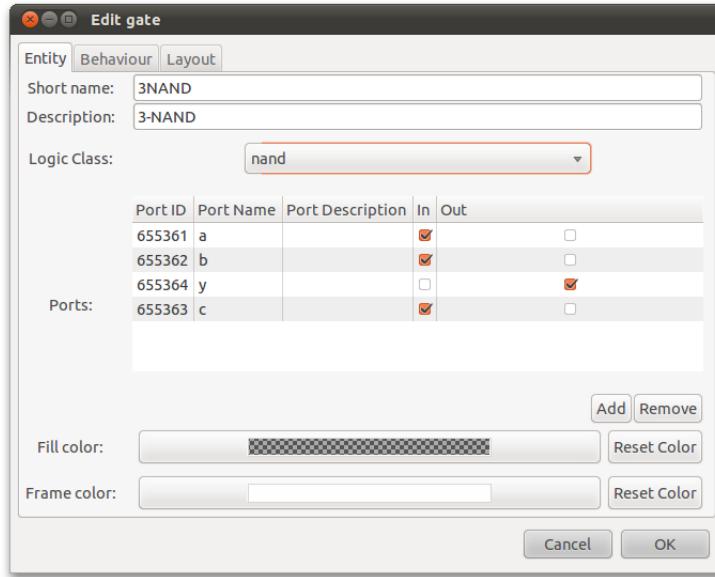


Figure A.6: Editing the Interfaces of a Standard Cell.

background and border colors for all the standard cell instances of the same type can be selected in the dialog box. This makes it possible to instantly recognize the type of an element based on the color of choice, such as flip-flops and XOR gates that are frequently encountered in cryptographic algorithms.

All entries in this dialog box are initially optional. Also, names need not be unique. It is nevertheless very useful to specify unique short names.

If the function of a given standard cell is determined (see section 3.6), this finding can be documented under the *behavior* tab. The behavior can be described as free text, in the form of truth tables, or even by means of hardware description languages. Furthermore, it is possible to store test cases, so-called test benches, for the purpose of describing the behavior.

If the field *Logic Class* is defined, it is possible, by pressing a button, to generate a code framework for the behavioral description and the specific test bench. It supports Verilog and VHDL and are supported. The code generator uses the interfaces that are specified under the tab *Entity* in the same dialog window. Specific ports, such as reset and clock ports are identified by common names (e.g. “rst”). More on this in Chapter 9.

The position of ports within standard cells is set in the main application window. To do this one clicks the right mouse button on the desired position within a standard cell instance, then selects

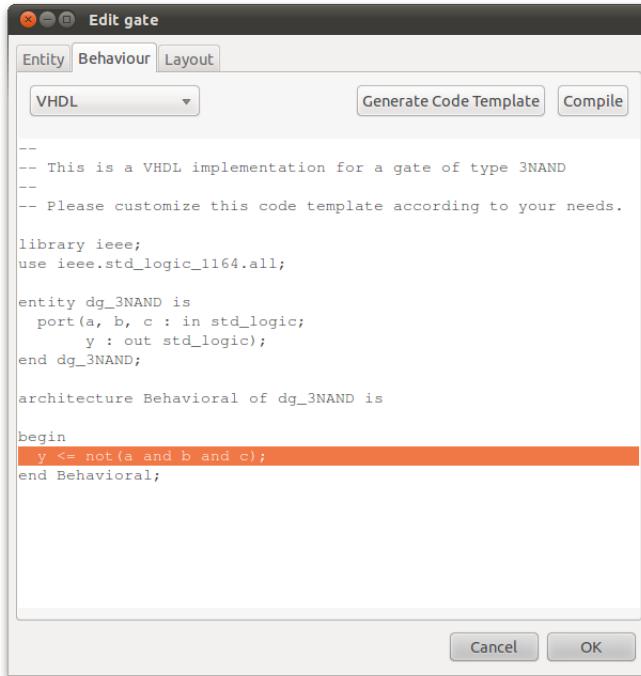
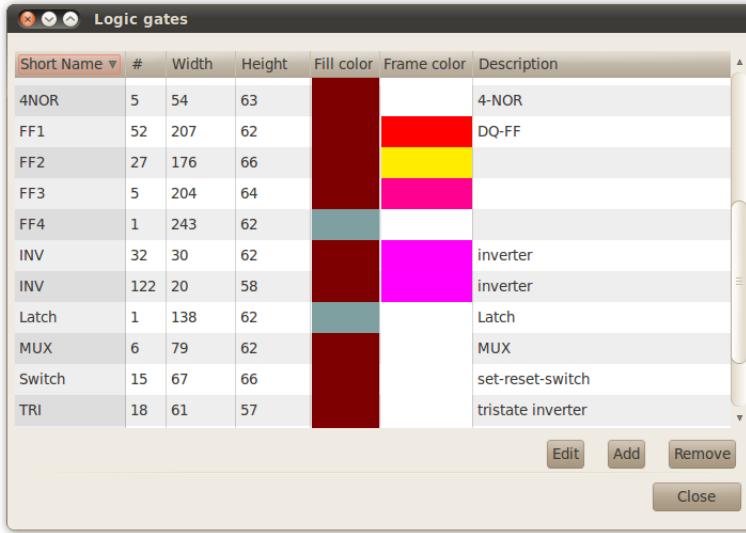


Figure A.7: Editing the Behavior of a Standard Cell

the item *Set port on gate* in the pop-up menu that appears. This opens another dialog box where all the ports of the chosen standard cell are listed. In the dialog box a port can now be chosen. The determination applies to all instances of the standard cell type. Definition and standard cell placement ports are separate steps. Ports can be rearranged at any time.

Once it has been established that a given area of the image represents a standard cell, after the process of describing the latter through the dialog has been completed, the same area of the image will be used as graphical representation of the layout of that standard cell type. The graphical representation may comprise several layers of the chip: The (first) transistor level and the levels M1 and M2. Furthermore, in the selected area, a first standard cell of the type will be placed. The graphical representation of the layout is stored in the standard cell definition and is independent of whether standard cells are actually placed in a(nother) project.

The list of the standard cells in a given project can be consulted in the *Gate Library* view (See Figure A.8).



The screenshot shows a software interface titled "Logic gates". It displays a table of standard cell definitions with the following columns: Short Name, #, Width, Height, Fill color, Frame color, and Description. The table contains 12 rows, each representing a different logic gate or cell type. The "Fill color" and "Frame color" columns show colored squares corresponding to the cell's visual representation. The "Description" column provides a brief description of each cell.

Short Name	#	Width	Height	Fill color	Frame color	Description
4NOR	5	54	63			4-NOR
FF1	52	207	62			DQ-FF
FF2	27	176	66			
FF3	5	204	64			
FF4	1	243	62			
INV	32	30	62			inverter
INV	122	20	58			inverter
Latch	1	138	62			Latch
MUX	6	79	62			MUX
Switch	15	67	66			set-reset-switch
TRI	18	61	57			tristate inverter

Figure A.8: The Gate Library, where all standard cell definitions are summarized.

## A.5 Finding Instances of Standard Cells

If standard cell types are defined in the respective gate library with the corresponding images as a template, it is possible to perform searches for instances of standard cells. In the graphic user interface a section of the image area in which to perform the search is marked with the mouse. If no search area restriction set, extending the search to the entire area.

Standard cell instances can be searched on different layers of the chip, wherein the level M1 is the best suitable for searching. The search is always performed on the layer that is currently displayed in the main application window. To each chip layer Degate associates its type information, i.e. whether the layer contains transistors or conductors. This enables the software to use the appropriate image material as a template for the search for each layer. Standard cells are represented in the graphical representation in all layers. In the logic model, they are saved on the first layer whose type is “metal.”

Under the menu item *Recognition* one can choose between three different methods: The normal *Template Matching* searches for standard cells in a chosen area of the image material; *Template Matching in Rows* resp. *Template Matching in Columns* are modified versions of the search algorithm, whereby the template is translated along the auxiliary grid lines. The latter two methods require that the auxiliary lines be situated either above the row or to the left of the columns contain-

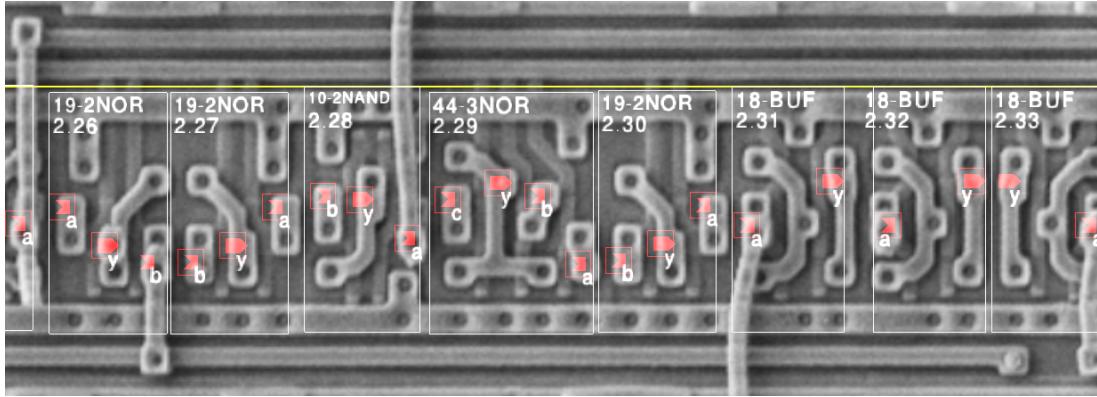


Figure A.9: Representation of Standard Cells in Degate

ing the cells. This requirement not only significantly reduces the search space, but also the number of false positives. Degate also takes into account the fact that standard cell instances often present themselves on the chip in a mirrored form. The search will then look for all possible orientations. The search can also be restricted to certain specific orientations, if desired.

If one of these menu items is invoked, a dialog box opens where one or more standard types of cells that are to be searched can be selected. In another dialog box, the parameters for the search process are determined. The search procedure is discussed in detail in Section 7.

Once a picture has been set as a template for a standard cell, it is first convenient to look for further instances for in a relatively small section of the whole image material. If the software has found a few instances of standard cells, they can be marked. Under the menu ► Gate → Use images of selected gates a function can then be invoked for merging image data from multiple instances of standard cells in a single picture (a *mixed image*), which is then used as a model for further searches.

The underlying problem is that a single template is only suitable as a representative. As a consequence of polishing disturbances arise regularly on the chip surface, e.g. areas that were eroded to varying degrees. For local searches it can thus be selected in each case a suitable local template. For a global search by means of a *mixed image* as a template will increase the detection rate, and the error rate decreases. The function to create mixed images can be applied to various instances of standard cells. The instances can be of different types of cells, and the function also takes into account various orientations.

Figure A.9 displays how standard cells are represented graphically in the software.

---

## A.6 Electrical and Design Rule Checks

The development of integrated circuits often pushes at the limit of what is technically feasible. The integration density is optimized for a target process in semiconductor technology: transistors, interconnects and vias are sized and placed in order to optimize space. One obvious reason is that more chips per wafer translate in a lower overall production cost. It is also possible to get more chips out of a wafer by shrinking the production process, but this also comes with increased costs per wafer. The choice of the semiconductor process is therefore a balance to be achieved between technical parameters (such as clock frequency of the chip) and the financial investment.

In order to timely detect errors, it is necessary to verify compliance with rules. This test is part of the layout verification and concerns layout data and electrical parameters. These tests are accordingly named “Design Rule Check” and “Electrical Rule Checks.” For instance, a “Design Rule Check” verifies that minimum distances be maintained between conductors. The actual rules are dependent on the semiconductor process. “Electrical rule checks” ensure that electrical parameters are not violated. For example, an output port of a standard cell can drive only a limited number of input ports of other standard cells. This parameter is called *fan-out* and is defined in the standard cell libraries.

In reverse engineering rule checks help to detect errors. The complexity of the rules is lower (than in chip development) as most of the rules need not be tested because the chip is (assumed to be) working.

Degate supports rule checks. Violations of rules can be viewed and edited by means of the GUI. Electrical Rule Checks recognize:

- Unconnected inputs of standard cells,
- Compounds of several output ports,
- Input ports which are not connected with any output port.

Design rule checks identify:

- Too closely spaced electrically connectable objects that are not connected to one another,
- Vias for which no counterpart in the neighboring layers exist.

Design rules in Degate are not limited solely to test the layout. In fact, they are useful to detect layout-related connection problems. They are ultimately a special form of Electrical Rules.

---

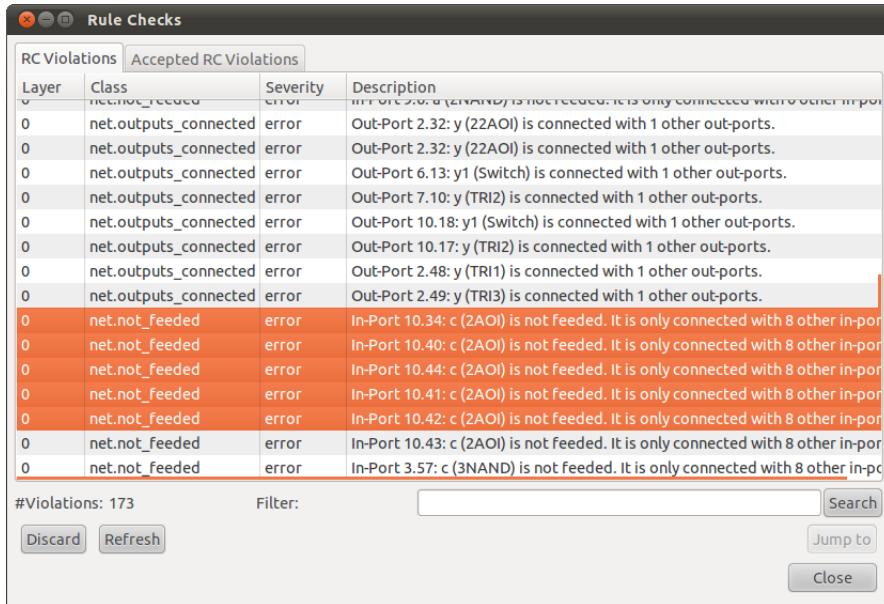


Figure A.10: Rule Check Dialog Box

To start the Rule-Check (RC) dialog select the menu ► Logic → Rule checks. This performs the rule checks and displays detected and potential problems in a table, as shown in Figure A.10. The *Severity* column indicates how serious is a rule violation, whether it is a warning or an error. Information on the type of problem is shown in the *Class* column. The *Description* column is an account of the error type and which logic model object is affected. When a single table row is selected, the button *Jump to* is activated, by which one can view the object in question in the main window. The list of violations can be searched. Under the table in the RC dialog there is a search field - it is used as a filter to show only the lines that contain the search phrase.

RC-violations for individual logic model objects can be disabled. While errors refer to any actual problems, warnings mean that Degate can not unambiguously determine the presence of an error. RC errors bugs should therefore never be ignored. Ignored RC violations are persistent and are stored in a separate XML file. On the tab *Accepted RC Violations* it is however possible to manually withdraw RC problem reports. Degate then asks whether a new RC-run should be executed. Manually removed rule violations are ignored automatically in the new run.

## A.7 Grouping Standard Cells

Modules are an organizing principle for forward circuit design and serve to create a hierarchical structure and enable reuse of functional and structural descriptions (see Chapter 9). Standard cell instances in Degate are grouped into modules - in order to keep together the results of the analysis of lists of circuits and to enable the export of descriptions of modules.

Each Degate project consists of a main module, in which initially all standard cell instances are grouped together. The user can add sub-modules and assign standard cell instances to these. This creates a hierarchical tree structure of Modules, that can be see in Figure A.11.

Module Hierarchy						
Module Name	Module Type	Gate Name	Gate Type	Module Port Name	Gate Port	Gate
main_module		3.40 : FF2	FF2	clk	3.40: clk (FF2)	3.40 : FF2
crypto		3.31 : FF2	FF2	ctrl_a	1.44: a (2AOI)	1.44 : 2AOI
mux81		2.39 : FF2	FF2	ctrl_b	1.44: b (2AOI)	1.44 : 2AOI
lfsr_a	lfsra	1.43 : FF2	FF2	ctrl_c	1.44: c (2AOI)	1.44 : 2AOI
lfsr_b	lfsrb	1.45 : XOR	XOR	mux_a	3.40: IQ (FF2)	3.40 : FF2
		1.44 : 2AOI	2AOI	mux_b	3.31: IQ (FF2)	3.31 : FF2
		2.37 : FF2	FF2	mux_c	2.38: IQ (FF2)	2.38 : FF2
		2.38 : FF2	FF2			
		2.40 : FF2	FF2			

Figure A.11: Assignment of Standard Cell Instances to Modules

## A.8 Further Functions

Annotations are a kind of bookmark that can be entered on a chip layer. They are useful to retrieve places or areas and their inscriptions.

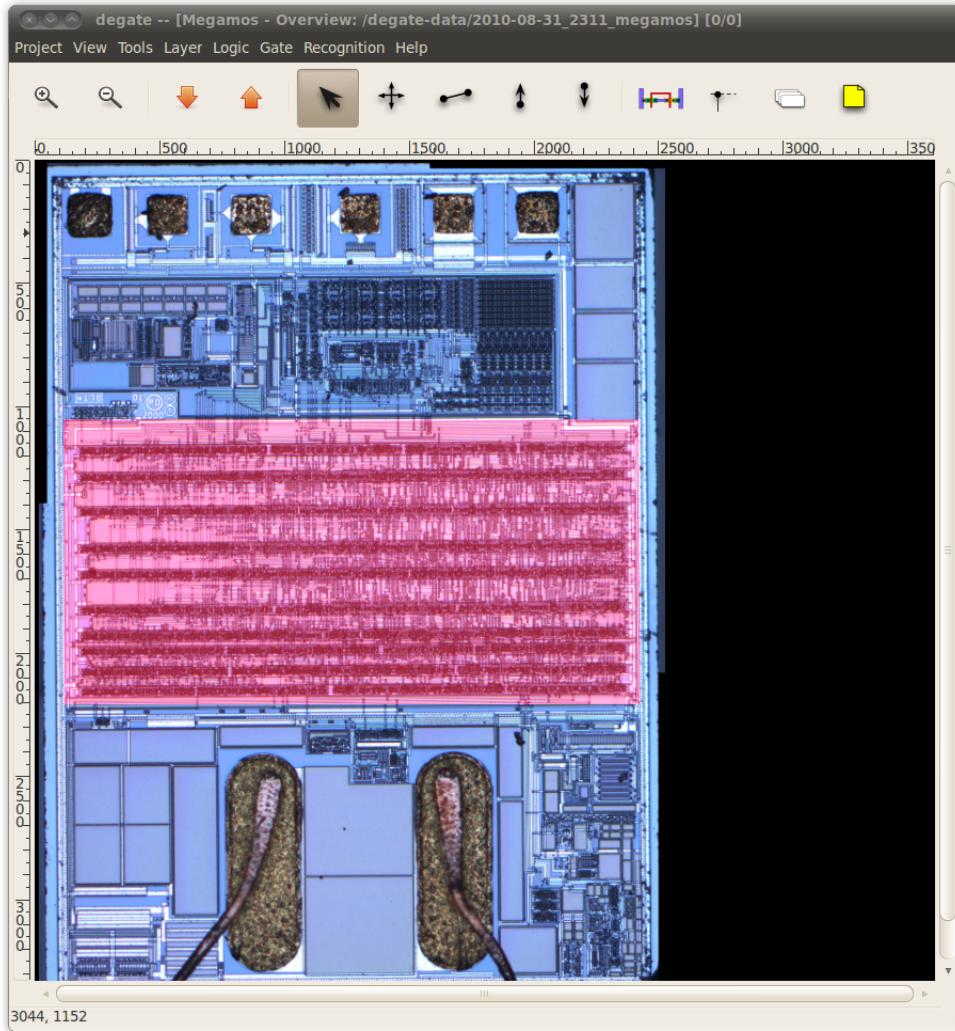


Figure A.12: Visualisation of Subprojects

Clicking on the toolbar or invoking the menu entry ► Logic → Annotation one can view a list of all annotations. This list allows to navigate directly to the annotations.

Pictures are often taken only from areas of the chip that are relevant for an analysis, for example, where one suspects that cryptographic functions are implemented or a detail of a memory bank in order to identify the type of memory. To keep track of which areas were photographed, sub-projects can be used. In the user interface these are shown as annotations. By double-clicking with the mouse, one changes the focus to a sub-project. The menu entry ► Project → Open parent project takes back to the parent project. Technically, they are independent Degate

---

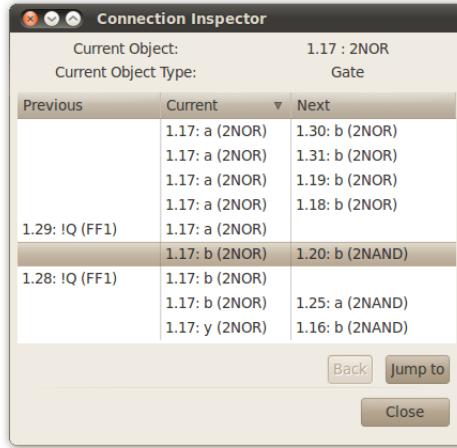


Figure A.13: The Connection Inspector

projects. Projects represent a separate container for images and objects in the respective parents. In each parent project the informations specific to a sub-project are not stored, but only where they are located inside the parent project and where the to find the corresponding project data in the file system. Figure A.12 shows how subprojects are represented in the main window.

Objects of the logic model, for example, placed standard cells, traces and annotations have names. For placed standard cells, it has proven to be useful to give them a name that encapsulates the position. For instance, the name “5.23” can be used for the 23<sup>rd</sup> gate in the fifth column. This makes it possible, by using the name to construct spatial relationships to other standard cell instances. Since functionally related standard cells are usually placed in adjacent positions one can heuristically infer that spatial relationship implies functional relationship. The action for the automatic naming by position is invoked by selecting one the menu items ► Gate → Generate names along rows and ► Gate → Generate names along columns.

The dialog box “Connection Inspector” is used to understand electrical connections in the logic model. If an object is clicked with the mouse, it will be shown in the middle column of the Connection Inspector. This is shown in Figure A.13, where for instance a NOR gate is selected. The NOR gate is named “1.17”, has two inputs *a* and *b* and an output *y*. These ports are listed in the middle column. In the left column outputs of standard cells are specified, which are electrically connected to inputs of the selected object (“fan in”). Here, these are negated outputs of two different flip-flops. Accordingly, in the right column inputs connected to output ports of the selected gate are recorded (“fan out”). Electrically connectable objects that have no input or outputs, such

as conductors, are shown both in the left and right columns.

The window has a button “Jump to” to go to electrically connected objects. The window remembers which element one is jumping from, so it is possible to jump back using the “back” button. The “Connection Inspector” is not modal, so it can remain open while the user is working int the main window.

Occasionally it is desirable to mark ports of standard cells according to their function with colors, for instance reset inputs always red and clock interfaces always green. The dialog “port Colors” provides the means to give project global color to inputs and outputs of standard cells. The assignment uses the port name.