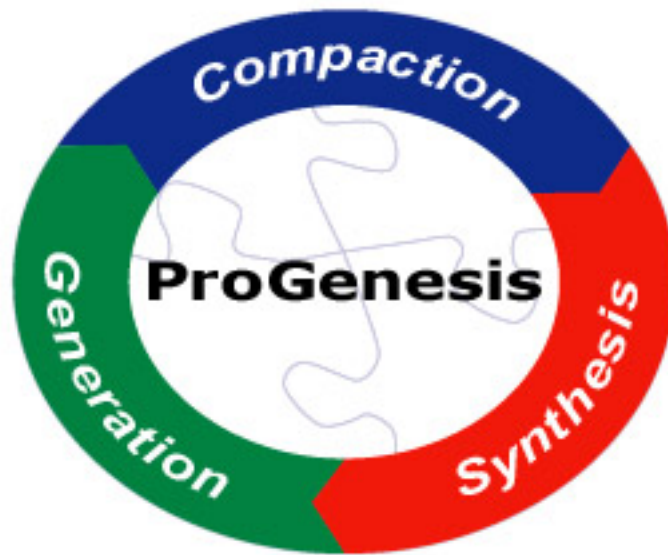


# Introduction To ProGenesis



**Prolific, Inc.**

## **Introduction To ProGenesis**

by Prolific, Inc.

Copyright © 1998-2002 by Prolific, Inc.

Introduction to Prolific Inc.'s ProGenesis™ standard cell creation software.

Every effort has been made to supply complete and accurate information. However, PROLIFIC, Inc. assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

No part of this publication may be stored in a retrieval system, transmitted, or reproduced, in any form or by any means, including but not limited to electronic, magnetic, mechanical, photocopying, recording, or otherwise, without the prior agreement and written permission of PROLIFIC, Inc.

ProGenesis™ is a trademark of Prolific, Inc.

# Table of Contents

<b>1. Getting Started With ProGenesis</b> .....	<b>1</b>
1.1. ProGenesis Tool Overview .....	1
1.1.1. ProTech .....	1
1.1.2. ProSpin .....	2
1.1.3. ProGen.....	2
1.2. ProGenesis Setup .....	2
1.3. Recommended Directory Structure.....	3
1.4. Running Demonstration Data .....	5
1.4.1. Generating Cell Layout From A Netlist.....	6
1.5. Customize The System .....	7
1.5.1. Setup Tools for the Library .....	7
1.5.2. Define the Requirements.....	7
<b>2. ProGenesis Tutorial</b> .....	<b>9</b>
2.1. Overview .....	9
2.1.1. ProGenesis Flow.....	9
2.1.2. ProGenesis Approach.....	10
2.2. ProGenesis Tutorial Guide.....	10
2.2.1. Prepare To Run Tutorial Data .....	10
2.2.2. Starting ProTech .....	11
2.2.3. Entering Setup Data.....	12
2.2.4. Starting ProSpin .....	16
2.2.5. Running ProSpin.....	17
2.2.6. Starting ProGen.....	20
2.2.7. Evaluating ProGen Output .....	20
2.3. Tutorial Data .....	21
2.3.1. Copy Tutorial Data.....	21
2.3.2. Configure Design Rules and Cell Template .....	21
2.3.3. Input Netlist.....	29
2.3.4. Create Layout .....	29



# Chapter 1. Getting Started With ProGenesis



## 1.1. ProGenesis Tool Overview

The basic ProGenesis tool flow is shown in Figure 1-1. ProGenesis is the suite of tools that generate standard cell layout from SPICE input. ProGenesis includes 3 main tools used to create layout from netlists: ProTech, ProSpin, and ProGen. Although other tools are included in the ProGenesis suite, these are the three we'll focus on to get started.

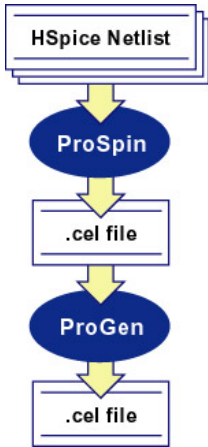


Figure 1-1. ProGenesis Flow

### 1.1.1. ProTech

ProTech is the tool used to configure fabrication technology specifications, design styles, and layer information. ProTech is used to enter or modify the following types of information:

- Design Options: Transistor layout and folding, compaction and routing options
- Design Rules: Mandatory and preferred rules specified by the fab
- Layer Data: Specify GDS layer data and text
- Cell Template: Cell height, well tie behavior, and rail and well sizes.

### 1.1.2. ProSpin

ProSpin reads in SPICE netlists describing individual cells and produces corresponding CEL files to be read by the ProGen tool. A CEL file specifies the generators that should be called to create the final layout data and contains cell-specific information such as transistor sizes and node names.



Figure 1-2. ProSpin Flow

### 1.1.3. ProGen

ProGen reads in CEL files and produces physical layout based on the cell characteristics, the generators specified, and the technology-specific information created using ProTech. After reading the CEL file, ProGen invokes the proper generators to produce a loose physical layout. ProGen then compacts this initial layout to produce a final cell that is as small as possible while conforming to all design rules and layout constraints.

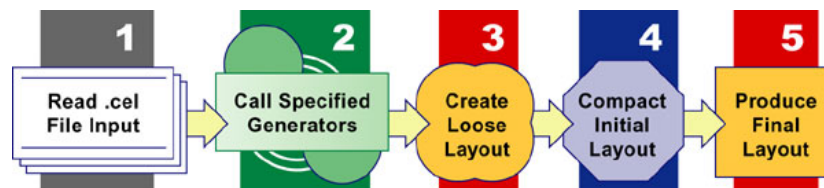


Figure 1-3. ProGen Flow

## 1.2. ProGenesis Setup

Once ProGenesis has been installed on the system, configuring the environment is easy: The PROLIFIC environment variable must point to the directory where the software has been installed.

For example, if the software were installed in `/usr/local/prolific/v4.0` on your system, then you could use the following command to set the environment using the Bourne shell:

**Example 1-1. Setting the PROLIFIC environment variable in the Bourne shell.**

```
sh% PROLIFIC=/usr/local/prolific/v4.0
sh% export PROLIFIC
```

For non-Bourne shells, use the syntax appropriate to the shell; for example, using `csh`, set the PROLIFIC environment variable this way:

**Example 1-2. Setting the PROLIFIC environment variable in the C shell.**

```
csh% setenv PROLIFIC /usr/local/prolific/v4.0
```

To check the environment variable, use the following command:

```
csh% echo $PROLIFIC
```

From now on, we'll refer to the PROLIFIC installation directory as `$PROLIFIC`. All of the tools reside in the `$PROLIFIC/bin` directory, so an easy way to run them is to add the `$PROLIFIC/bin` directory to the path. It is most convenient to do this in the shell configuration files; for example, in the `csh` configuration file, `.cshrc`, the following command can be added:

**Example 1-3. Adding the Prolific tools directory to the path in the `.cshrc` file.**

```
set path = (
  $PROLIFIC/bin \
  $path \
)
```

## 1.3. Recommended Directory Structure

While you are free to organize your data as you wish, PROLIFIC has a recommended structure for those who want to use it. As shown in the diagram below, PROLIFIC suggests creating a separate tree for each internal or external customer, with separate subdirectories for each project created for that customer. For those

## Chapter 1. Getting Started With ProGenesis

who have minimal input information, the net directory will suffice for the base netlists. The incoming directory can be used for more detailed customer data specifications, such as examples and customer documentation. Other standard directories include:

psc

contains cells created using ProSticks

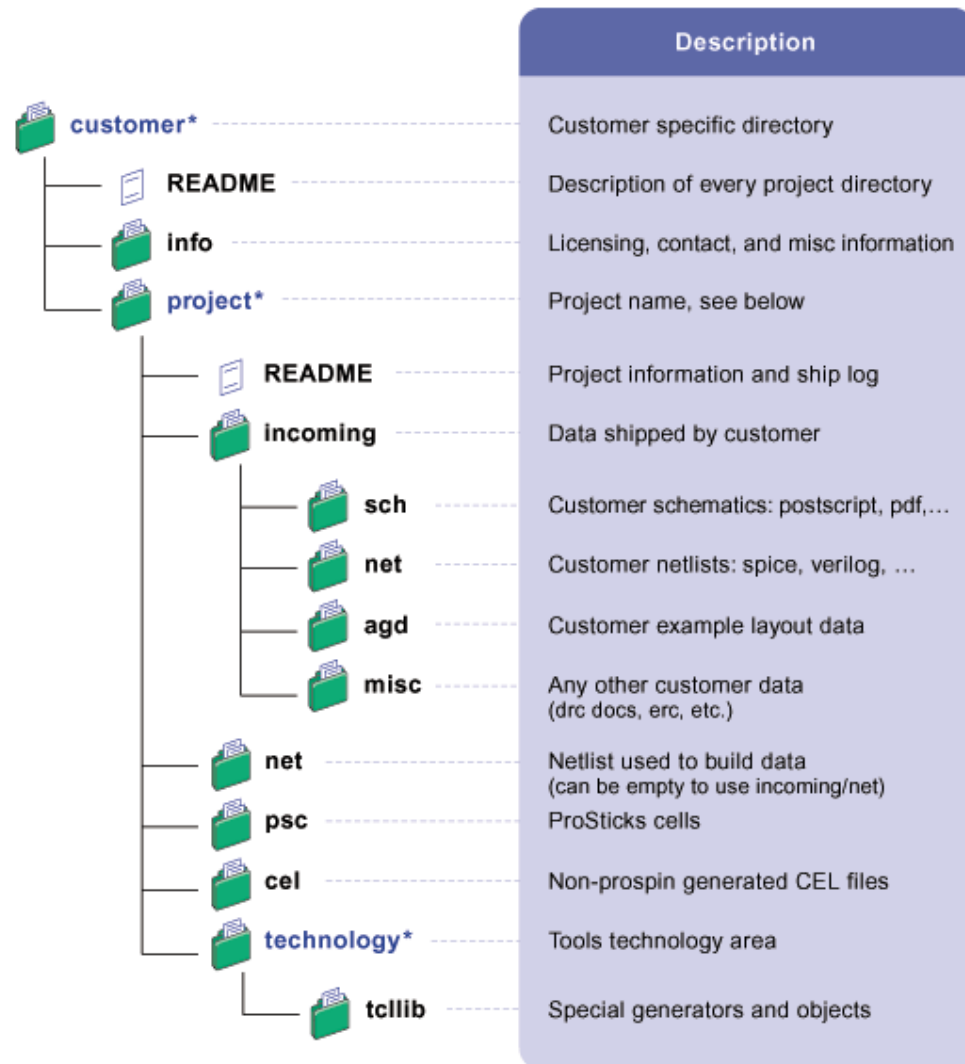
cel

contains cell-specific CEL files for cells not created by ProSpin

technology

for each fabrication technology, a uniquely-named directory





#### Special Definitions

<b>customer</b>	= Customer name
<b>project</b>	= proj<int><int> [_desc] Customer project ID
<b>&lt;int&gt;</b>	= [0-9]
<b>technology</b>	= technology name

Figure 1-4. Suggested Directory Structure

## 1.4. Running Demonstration Data

PROLIFIC provides a demonstration area which contains everything you need to run the ProGenesis tools. All of the examples in this section use data from this demonstration area:

```
$PROLIFIC/demo
```

The demo area contains the following types of data:

- demonstration CMOS technology
- cell SPICE netlists
- sample layout AGD (ASCII GDS, the output from ProGen) data
- sample ProSticks layout, generators, cells, and output AGD data

### 1.4.1. Generating Cell Layout From A Netlist

This section describes how to generate an inverter cell from a SPICE netlist. This process involves three steps:

1. Create or move to a working directory
2. Run ProSpin to convert the SPICE netlist to a CEL file
3. Run ProGen to generate the cell and compact it

#### 1.4.1.1. Step 1: Move to a working directory

The ProGenesis flow results in many files being created, so it is wise to run the tools from an appropriate working directory. We suggest creating a parent directory for each library project, with subdirectories for netlist input, CEL files, and process technology information. See the Recommended Directory Structure section earlier in this document for details.

#### 1.4.1.2. Step 2: Use ProSpin to Convert Netlist to CEL File

In this step, we run ProSpin from the command line to create a cell description file named `inv.cel` from a SPICE netlist named `inv.sp`. Use the following command:

```
ssh% prospin -l inv $PROLIFIC/demo/spice/inv.sp
```

This should produce the `inv.cel` file in your working directory. ProSpin automatically creates CEL files to match each sub-circuit name requested. For more information, consult the ProSpin Reference Guide.

#### 1.4.1.3. Step 3: Use ProGen to Generate and Compact the Cell.

Now that there's a CEL file corresponding to the inverter cell, we run ProGen to create the cell layout:

```
ssh% progen -t $PROLIFIC/demo/tech/demo.db -i inv.cel -o  
inv.agd
```

This command produces the `inv.agd` cell file, which contains the layout in AGD format. For more information on ProGen, consult the ProGen Reference Guide.

#### 1.4.1.4. Step 4: Viewing AGD Data

While this step isn't necessary for generating cell layout, it is useful for viewing the layout created in the first three steps. The ProGenesis tools produce AGD (ASCII-GDS) cell data. AGD is an ASCII representation of the standard GDSII data file format. PROLIFIC provides an AGD data file viewer called `ProView`.

```
csch% proview -t $PROLIFIC/demo/tech/proview.tcl inv.agd
```

This opens the `ProView` viewer from X Windows and loads the AGD file `inv.agd`. Further information on options available in `ProView` is available in the `ProView Reference Guide`.

#### 1.4.1.5. Step 5: Converting Between AGD and GDS

Moving the data from ASCII GDS to a binary GDS file is easy. PROLIFIC provides two easy-to-use utilities for converting layout between AGD and GDS formats. To convert AGD data to GDS, use `a2gds`:

```
csch% a2gds < inv.agd > inv.gds
```

To convert GDS data to AGD, use `gds2a`

```
csch% gds2a < inv.gds > inv.agd
```

## 1.5. Customize The System

This section of the documentation describes how to set up the system to meet specific requirements.

### 1.5.1. Setup Tools for the Library

The preceding sections have highlighted the configuration required for getting the ProGenesis tools running. To customize an environment to make use of the ProGenesis tools, be sure to add the PROLIFIC variable to the environment and the `$PROLIFIC/bin` directory to the path each time the tool user logs on.

Also, create a directory structure to contain the various source files and derived data for the project.

### 1.5.2. Define the Requirements

The tools will produce a standard cell library based on the technology setup. Knowing what the results should look like will be very helpful in setting up the technology files. The following is a list of items needed to develop a good technology definition for a library:

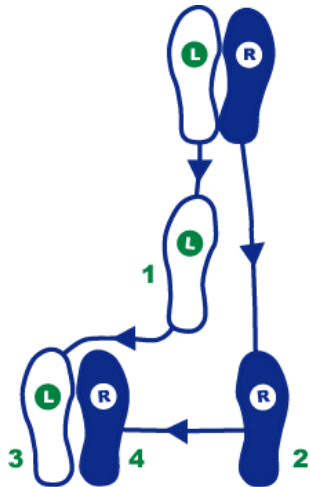
- design rules

## *Chapter 1. Getting Started With ProGenesis*

- general technology characteristics (layers, routing)
- general cell requirements (height, grid, rails)
- types of cells in library (AOI, flops, latches)
- extras (diodes, ties)

As with most things in life, it is easiest to tackle ProGenesis one step at a time. When configuring the system for your library the first time, don't try to implement every last option. Instead, focus on the broad, general features first and then begin fine-tuning the features after the general settings are working.

## Chapter 2. ProGenesis Tutorial



### 2.1. Overview

The ProGenesis suite allows designers to quickly and accurately create optimized standard cells. ProGenesis includes the tools shown in Table 2-1:

Table 2-1. ProGenesis Tools

Tool	Description
ProTech	Design rule and configuration tool
ProSpin	Netlist analysis tool
ProGen	Cell creation and compaction tool

#### 2.1.1. ProGenesis Flow

ProGenesis can create and manage any number of cells, and can handle cells as part of a library or individually. ProGenesis requires the inputs shown in Table 2-2 to create GDS-II output:

Table 2-2. ProGenesis Input

Input	Description
Netlist	Single, multiple, or hierarchical HSpice netlist
Design rules	From design rule document
Cell template definition	Design-goal driven, user-defined configuration

The cell creation process follows these steps:

1. Define setup information in ProTech: design rules and cell template definitions.

---

Input	Enter design rules and cell architecture settings
Output	A database (.db) file containing design rules, cell template, and library option data

---

2. Use ProSpin to analyze netlists and match their contents to cell generators.

---

Input	HSpice netlists
Output	.cel files containing the generator calls corresponding to each cell

---

3. Run ProGen to create and compact cells.

---

Input	Output from the previous steps (.cel files and .db file)
Output	GDS-II layout

---

## 2.1.2. ProGenesis Approach

ProGenesis already recognizes the basic hierarchical building blocks in a cell, such as NAND, AOI, and TRI-INV. These blocks are represented within the ProGenesis flow as "generators," which can include descriptions of transistor placement, routing configurations, and other layout details.

In general, more layout information contained in a generator results in more predictable GDS-II output. However, it is also possible to include too much information in a generator, which can overconstrain the cell creation process, ruling out desirable solutions. Walk through the Section 2.2 to see how generators are used.

## 2.2. ProGenesis Tutorial Guide

This tutorial will walk you through the process of running the ProGenesis tools for the first time. Sample tutorial data should be included with the ProGenesis tools; the data are also available in Section 2.3.

### 2.2.1. Prepare To Run Tutorial Data

Ensure that the environment and tools are ready:

- Create a working directory you can use without worrying:

```
% mkdir temp
```

- Create a subdirectory structure for data and output:

```
% cd temp
```

```
% mkdir agd cel net  
tech
```

- Check that the environment variable is set correctly:

```
% echo $PROLIFIC
```

**Setting the Environment Variable:** If the PROLIFIC environment variable is not set, or points to the wrong directory, follow the setup instructions in the Getting Started guide.

- Check that ProGen runs properly:

```
% progen -h
```

- Make a local work directory and copy the tutorial data there:

```
% mkdir work  
% cp $PROLIFIC/tutorial work
```

### 2.2.2. Starting ProTech

Start ProTech from the command line:

```
% protech
```

The ProTech graphical interface, like the one in Figure 2-1 will appear on the screen. Its tabbed region identifies the different categories of configuration information, as shown in Table 2-3:

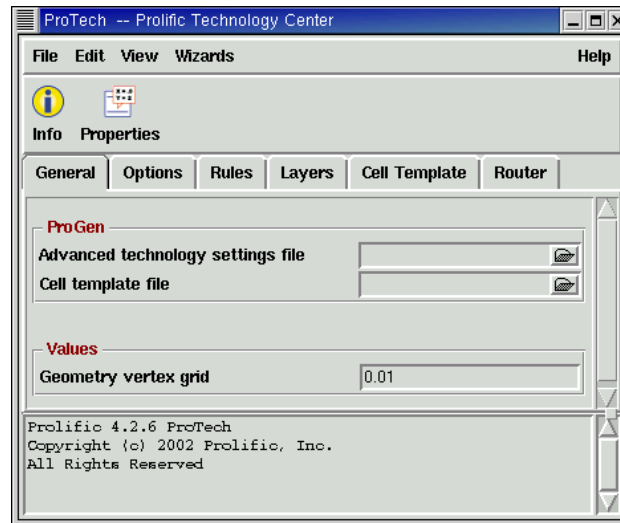


Figure 2-1. ProTech Interface

Table 2-3. ProTech Tabs

Tab	Description
<i>General</i>	Set manufacturing vertex grid. Set path to advanced settings file
<i>Options</i>	Configure library-wide options and layout style preferences
<i>Rules</i>	Enter design rules
<i>Layer</i>	Define layer-to-GDS mappings
<i>Cell Template</i>	Select cell template style
<i>Router</i>	Calibrates router for cell template definition

### 2.2.3. Entering Setup Data

The first time you use ProTech, follow these steps:


1. Click on the *General* tab. In the *Geometry vertex grid* field, enter the manufacturing grid size in microns. For this tutorial, enter **0.005**.
2. You can skip the other *General* settings and the *Options* settings. When you initially create cells, you do not have to alter these settings, but on subsequent iterations they may help you to tailor the layout to your needs and preferences.
3. Click on the *Rules* tab to enter design rules from a design rule document. For this tutorial, enter the values shown in Table 2-4. If you want to know more about a particular design rule entry, click on the  button and then click on the name of any rule for a description and illustration of that rule.



Table 2-4. Tutorial Design Rule Settings

<b>Rule Name</b>	<b>Value to Enter</b>
Min N well width	0.62
N well overlap N+ diffusion tie	0.17
N well overlap P+ diffusion	0.22
Minimum P+ implant width	0.24
P+ implant space to N+ diffusion	0.13
P+ implant space to N+ diffusion gate	0.22
P+ implant space to N+ diffusion tie	0.02
P+ implant overlap P+ diffusion	0.13
P+ implant overlap P+ diffusion gate	0.22
P+ implant overlap P+ diffusion tie	0.02
Minimum diffusion width	0.11
Minimum diffusion spacing	0.14
Diffusion overlap of contact	0.05
Diffusion extension from gate	0.16
Minimum width butted diffusions	0.11
N+ Diffusion space to N well	0.22
Minimum poly width	0.1
Minimum poly spacing	0.14
Minimum poly overlap of contact	0.03 0.05 0.03 0.05
Minimum poly space to diffusion	0.05
Poly-gate endcap extension	0.16
Minimum transistor gate width	0.12
Minimum poly-to-poly gate spacing	0.15
Minimum transistor gate length	0.1
Minimum contact width	0.12
Minimum contact spacing	0.14
Minimum diffusion-contact to gate spacing	0.08
Minimum poly-contact to gate spacing	0.1
Minimum Metal-1 width	0.12
Minimum Metal-1 spacing	0.12
Metal-1 overlap diffusion-contact	0.0 0.05 0.0 0.05

Rule Name	Value to Enter
Metal-1 overlap poly-contact	0.0 0.025 0.0 0.025
Metal-1 overlap Via-1	0.005 0.05 0.005 0.05
Metal-1 minimum area	0.059
Minimum Via-1 width	0.13
Minimum Via-1 spacing	0.15
Minimum Metal-2 width	0.14
Minimum Metal-2 spacing	0.14
Metal-2 overlap Via-1	0.005 0.05 0.005 0.05
Metal-2 minimum area	0.07

4. Now click on the *Layers* tab, where there are two regions containing data used to generate mappings between GDS-II output numbers and user-defined layer names, and between the user-defined layers and ProGenesis' internal layers.

Layers Section	Description
ProGen Layers	Match the layers from the pull-down menus to ProGen's system layers listed in the left-hand column. The layers in the pull-down menus are defined in the ProView Layers section.
ProView Layers	Assign or alter output GDS numbers and layer colors as needed. Be sure each layer is assigned to a unique GDS number.

For the tutorial, set the layers as shown in Table 2-5 and Table 2-6. Because the ProGen layers are set to match the ProView layers, start by configuring the ProView layers.

- 4.a. Start by adding ProView layers that are missing. Click on the **Add ProView Layer** button, enter the name of one layer to be added, and click **ok**. The tutorial layers that need to be added are **diffusion, via 2, metal 3**.
- 4.b. Now delete any ProView layers that are not used. Click on the **Delete ProView Layer** button, select the name of one layer to be removed, and click **ok**. The tutorial layers that can be removed are **n+ deep implant, p+ deep implant**.
- 4.c. Set the GDS layer for each of the ProView layers that remain. Simply click in the *GDS Layer* field and set the layer number as necessary. For the tutorial, refer to the layer numbers in Table 2-6.
- 4.d. Set the display properties for each layer:  
Color/Pattern

Show Fill Pattern?  
 Show Outline?  
 Show Layer?

Click on the color bar to alter the display color and hash pattern for each layer, and using the checkboxes under *Display Fill* and *Display Outline*, set each layer to be displayed with a fill pattern, an outline, or both. Add or remove layers from the display (but not the GDS-II output) by toggling the *Display View* field. For the tutorial, match the fill and display settings shown in Table 2-6.

4.e. If you wish, alter the order of the ProView layers by clicking on the **Change ProView Layer Order** button. The current layer list can be sorted as you like by clicking on a layer name and dragging it to a new location.

4.f. Once the ProView layers are configured, set the ProGen layers.

Start by matching the ProGen layers to the ProView layers. Click on the drop-down list under *ProView Data Layer* to set the ProView layer for each ProGen layer. For the tutorial, set n+ diffusion and p+ diffusion to **diffusion**.

4.g. Set the ProView text layer for each of the ProGen layers by clicking on the drop-down list under *ProView Text Layer* and selecting the desired layer name. For the tutorial, set metal 1 to use **metal 1** and set metal 2 to use **metal 2**.

4.h. To exclude a layer from being output, deselect its checkbox in the *Output Layer* column. For the tutorial, leave all the active layers' boxes selected.

**Table 2-5. Tutorial ProGen Layer Settings**

ProGen Layer	ProView Data Layer	ProView Text Layer	Output Layer?
cell border	cell border	text	<input checked="" type="checkbox"/>
n-well	n-well	text	<input checked="" type="checkbox"/>
n+ implant	n+ implant	text	<input checked="" type="checkbox"/>
p+ implant	p+ implant	text	<input checked="" type="checkbox"/>
n+ deep implant			
p+ deep implant			
n+ diffusion	<b>diffusion</b>	text	<input checked="" type="checkbox"/>
p+ diffusion	<b>diffusion</b>	text	<input checked="" type="checkbox"/>
poly	poly	text	<input checked="" type="checkbox"/>
contact	contact	text	<input checked="" type="checkbox"/>

ProGen Layer	ProView Data Layer	ProView Text Layer	Output Layer?
metal 1	metal 1	metal 1	✓
via 1	via 1	text	✓
metal 2	metal 2	metal 2	✓
critical path	critical path	text	✓

Table 2-6. Tutorial ProView Layer Settings

ProView Layer	GDS Layer	Display Fill?	Display Outline?
cell border	235		✓
n-well	3		✓
diffusion	6	✓	
n+ implant	26		✓
p+ implant	25		✓
poly	17	✓	
contact	30		✓
metal 1	31	✓	
via 1	51	✓	✓
metal 2	32	✓	
via 2	52	✓	✓
metal 3	33	✓	✓
text	900		✓
critical path	999		✓

The layer setup process allows you to assign more than one internal ProGen layer to the same GDS number. For example, there is more than one ProGen layer for diffusion. But if the process has only one layer for diffusion, first define that layer in the ProView Layers section. Then in the ProGen Layers section, choose that same layer from the pull-down menu for each of the ProGen diffusion layers.

5. Click on the *Cell Template* tab. Set the characteristics that apply to all cells here, such as cell height, routing grid, N-well position, and power rail dimensions.
6. Click on the *Router* tab, which is used to restrict the router to use routing that will fit the cell height. Click on the **Run parameter generation script** button only after completing the configuration on the *Rules* and *Cell Template* tabs.
7. Save the configuration using the **File** → **Save As ...** menu item.

## 2.2.4. Starting ProSpin

Start ProSpin from the command line:

```
% prospin -g
```

A graphical interface like the one shown in Figure 2-2 will appear on the screen.

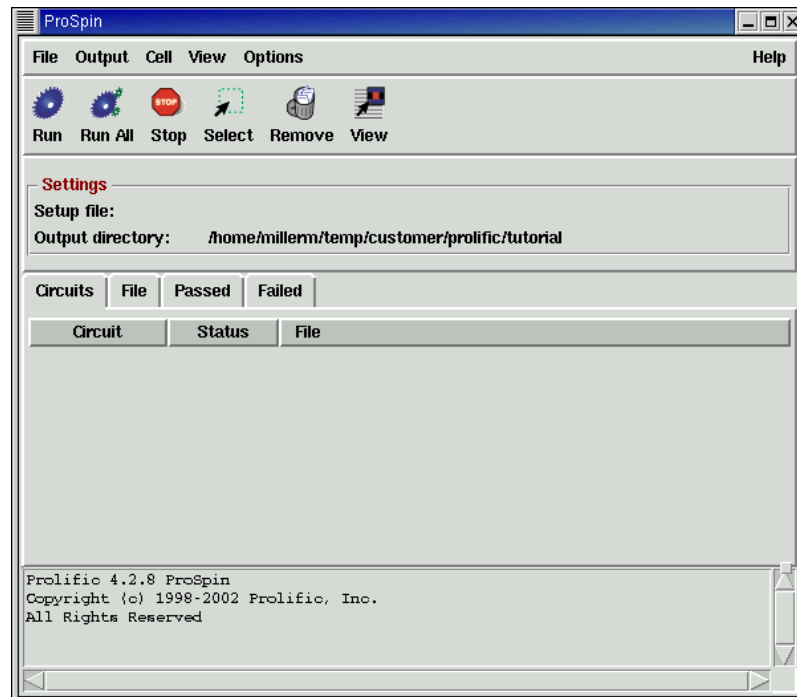


Figure 2-2. ProSpin Interface

## 2.2.5. Running ProSpin

ProSpin identifies the structures inside HSpice netlists and maps them to appropriate generators. Because netlists have many different styles, consider the following when starting to use ProSpin:

- Are transistor lengths and widths scaled properly? See Section 2.2.5.1 for more details.
- Are ProSticks generators referenced? See Section 2.2.5.2 for more details.
- Are names correctly specified to match power and ground nodes? See Section 2.2.5.3 for more details.
- Are ProPPR route settings correct? See Section 2.2.5.4 for more details.

### 2.2.5.1. Scaling Transistors

If the transistors need to be scaled by width or gate length, e.g., when porting netlists from an older technology, place the following commands in `prospin.tcl` to get linear scaling:

```
prospin_set option width_ratio_pfet x
prospin_set option width_ratio_nfet x
prospin_set option length_ratio_pfet x
prospin_set option length_ratio_nfet x
```

Set `x` equal to the scaling value:

$$\text{width}_{\text{desired}} = x * \text{width}_{\text{netlist}}$$

Setting these options allows the use of different scaling factors for p and n transistors.

### 2.2.5.2. Referencing ProSticks Generators

ProSticks is a layout editor that can produce generators. To use one of these generators in place of the default system generator, add these lines to `prospin.tcl`:

```
prospin_source generator.tcl

prospin_set option search_list {cellname} [list \
  [generatorproc_pattern_map]
]
```

Use the following values:

*generator.tcl*

The name of the Tcl file containing the generator

*cellname*

The cell name, which matches the subcircuit name from the netlist

*generatorproc*

The procedure name of the ProSticks generator

In most cases, all three values can be the same - use the `cellname` for all three variables. If you want to use a ProSticks generator as a subgenerator rather than a generator for the entire cell, add the following line to the end of the list:

```
[prospin_library basic default_list]
```

### 2.2.5.3. Specifying Power and Ground Node Names

If your netlist has a name for the power and ground rails that ProSpin doesn't recognize, use the following code in `prospin.tcl`:

```
prospin_set node_match ground {
  {pattern}
}

prospin_set nod_match power {
  {pattern}
}
```

This addition will perform a regular expression match for the node.

### 2.2.5.4. ProPPR Router Settings

To enable the ProPPR router, enable it in ProSpin by adding the following to `prospin.tcl`:

```
prospin_set generator options "^cellname$" {
  {proppr_route 1}
}
```

In addition to enabling the router, these are the additional commands that can control the behavior of the ProPPR router.

#### 2.2.5.4.1. Setting expected and acceptable costs

Set these when the router runs and finds a solution, but does not finish. Expected cost sets an upper bound on the router's solution: it will not select a solution with a cost greater than the expected cost. Acceptable cost is the cost where ProPPR will stop looking for solutions; the default is an exhaustive search so by default, the acceptable cost is set to 0.

```
prospin_set progen options "^cellname$" {
  pro_set global proppr_acceptable_cost cost
  pro_set global proppr_expected_cost cost
}
```

#### 2.2.5.4.2. Using joggedm1 and joggedpoly

For cases where the cell misses the cell height, or where outer channel m1 routes affect diffusion contact strapping, it may help to use `joggedm1` and `joggedpoly`.

```
prospin_set progen options "^cellname$" {
  pro_set global proppr_horz_layernames {poly joggedm1}
}
```

### 2.2.5.5. Using ProSpin

Follow these steps to run ProSpin:

1. Source the technology setup file, if applicable, and set output directory where .cel files will be placed. Use the **Options**→**Options...** menu item. For the tutorial, set the
2. Load the HSpice netlists using the **File**→**Load Spice ...** menu item.  
Select the netlist file or files to use, and click **OK**. The subcircuit names will appear in the main window along with the directory path of the netlist file being sourced.
3. Run ProSpin to map netlist functions to generators. To run ProSpin on all cells, click **Run All**. To run ProSpin on a subset of the imported cells, and click **Run**.

After creating the .cel files, select one of the cells and click **View** to examine the content of its .cel file. The .cel file contains calls to the generators ProGen will use. Each generator call is denoted by `flag_call`. While generator interfaces differ somewhat, the calls include: transistor widths, inputs, outputs, transistor names, connectivity.

View the contents of the .cel file to verify that a user-specified generator call (e.g., a generator created in ProSticks) is actually being called for the cell.

### 2.2.6. Starting ProGen

ProGen uses the technology database file and the .cel file to create and compact the cell layout. When ProGen runs on a cell, it first creates a fully-connected but loose topology, then compacts the cell according to the design rules and options in the database file.

ProGen runs from the command line:

```
% progen -t techfile -i  
celfile -o outputfile >&  
logfile
```

### 2.2.7. Evaluating ProGen Output

Once ProGen has created the output file, review the results using ProView. Use the following command to open all the output files in the current directory:

```
% proview -t techfile *.agd
```

Verify that the options that were set are reflected in the layout, and that the density meets expectations. If a cell appears wider than expected, it is possible to alter the



compaction order to reduce the cell's size, or to analyze the critical path to better understand the situation.

## 2.3. Tutorial Data

The information contained in this section can be used to run the sample cells without the hand-holding of the Section 2.2. You can make sure you understand the process of creating a library using these few cells:

- AND2
- LATCH
- MUX2
- XOR2

Following the flow described in Section 2.1.1, use the tools introduced there to create the layout:

- ProTech
- ProSpin
- ProGen

### 2.3.1. Copy Tutorial Data

Copy the tutorial directory to a local area:

```
% cp $PROLIFIC/tutorial
  foodir
```

### 2.3.2. Configure Design Rules and Cell Template

The tutorial area's directory structure is shown below:

```
tutorial
  |-- agd          AGD output from ProGen
  |-- cel          .cel files for use by
ProGen
  |-- net          HSpice netlists
  |-- tech         Technology database file and special option files
```

The design rule document for the tutorial is shown in Section 2.3.2.1.

### 2.3.2.1. Tutorial Design Rules

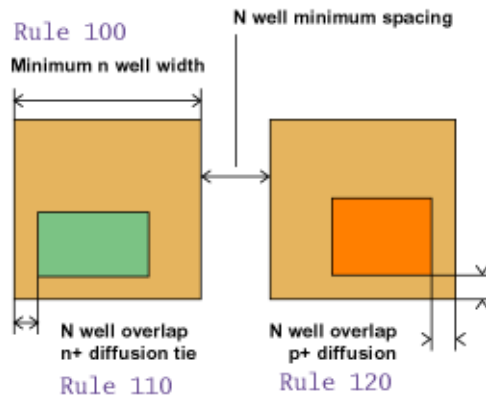


Figure 2-3. Well Rules

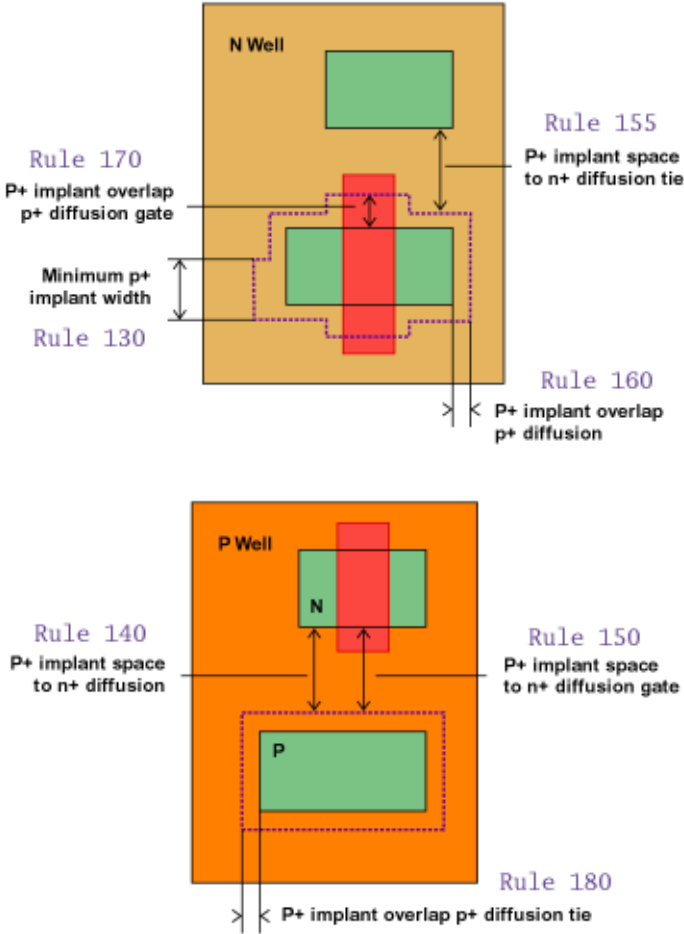


Figure 2-4. Implant Rules

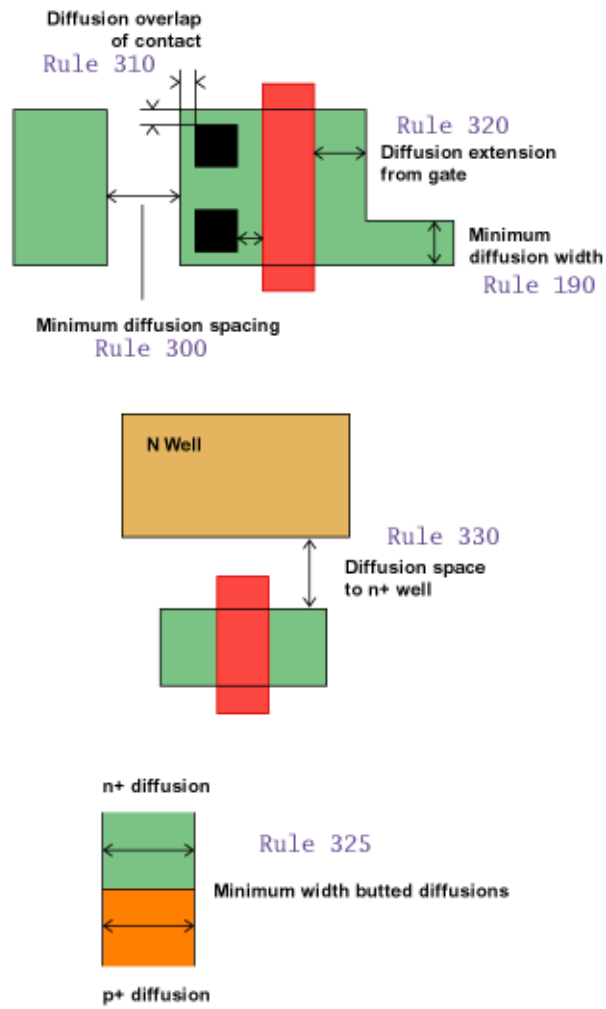


Figure 2-5. Diffusion Rules

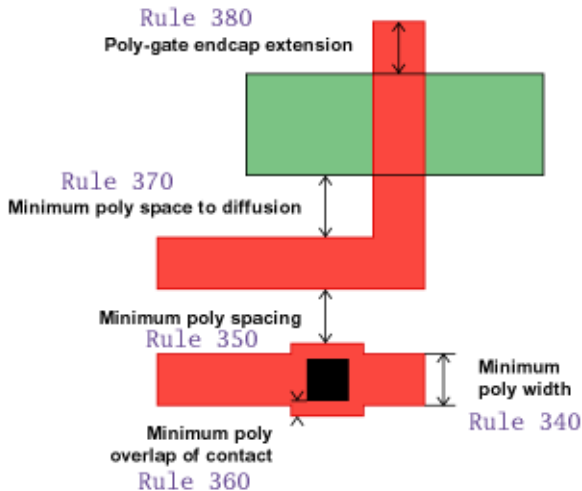


Figure 2-6. Poly Rules

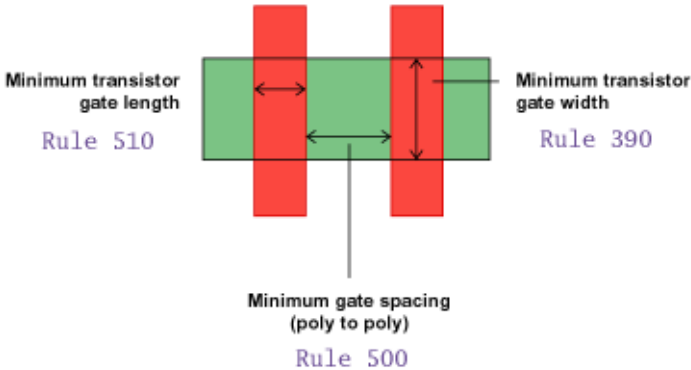


Figure 2-7. Gate Rules

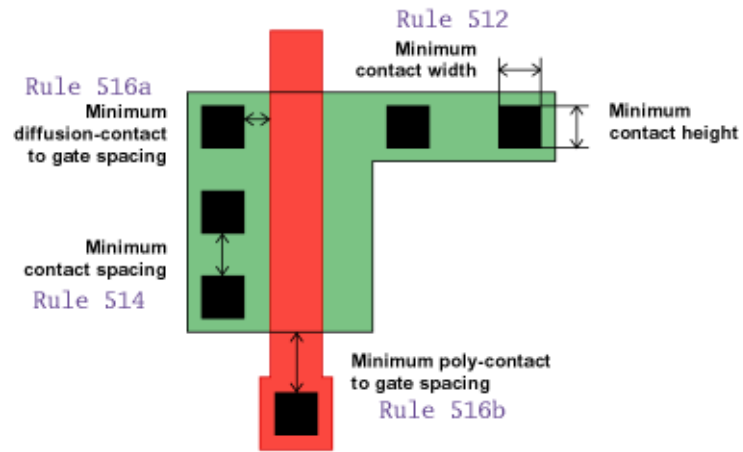


Figure 2-8. Contact Rules

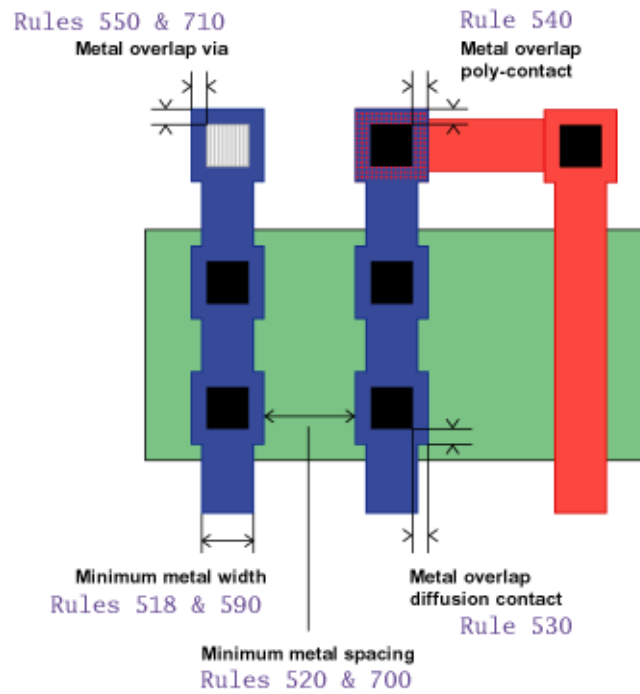


Figure 2-9. Metal Rules

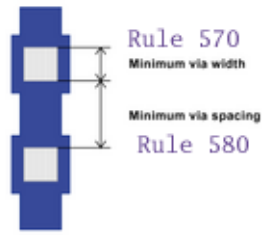


Figure 2-10. Via-1 Rules

Rule	Description	Value
100	Min N well width	0.62
110	N well overlap N+ diffusion tie	0.17
120	N well overlap P+ diffusion	0.22
130	Minimum P+ implant width	0.24
140	P+ implant space to N+ diffusion	0.13
150	P+ implant space to N+ diffusion gate	0.22
155	P+ implant space to N+ diffusion tie	0.02
160	P+ implant overlap P+ diffusion	0.13
170	P+ implant overlap P+ diffusion gate	0.22
180	P+ implant overlap P+ diffusion tie	0.02
190	Minimum diffusion width	0.11
300	Minimum diffusion spacing	0.14
310	Diffusion overlap of contact	0.05
320	Diffusion extension from gate	0.16
325	Minimum width butted diffusions	0.11
330	N+ Diffusion space to N well	0.22
340	Minimum poly width	0.1
350	Minimum poly spacing	0.14
360	Minimum poly overlap of contact	0.03 0.05 0.03 0.05
370	Minimum poly space to diffusion	0.05
380	Poly-gate endcap extension	0.16
390	Minimum transistor gate width	0.12
500	Minimum poly-to-poly gate spacing	0.15

<b>Rule</b>	<b>Description</b>	<b>Value</b>
510	Minimum transistor gate length	0.1
512	Minimum contact width	0.12
514	Minimum contact spacing	0.14
516a	Minimum diffusion-contact to gate spacing	0.08
516b	Minimum poly-contact to gate spacing	0.1
518	Minimum Metal-1 width	0.12
520	Minimum Metal-1 spacing	0.12
530	Metal-1 overlap diffusion-contact	0.0 0.05 0.0 0.05
540	Metal-1 overlap poly-contact	0.0 0.025 0.0 0.025
550	Metal-1 overlap Via-1	0.005 0.05 0.005 0.05
560	Metal-1 minimum area	0.059
570	Minimum Via-1 width	0.13
580	Minimum Via-1 spacing	0.15
590	Minimum Metal-2 width	0.14
700	Minimum Metal-2 spacing	0.14
710	Metal-2 overlap Via-1	0.005 0.05 0.005 0.05
720	Metal-2 minimum area	0.07

From the tech directory, follow the instructions in Section 2.2.3 to complete the rules and layers sections using the information in Section 2.3.2.1. For the data on the **Cell Template** page, enter the information in Section 2.3.2.2.

### 2.3.2.2. Tutorial Cell Template

<b>Option</b>	<b>Value</b>
Cell Template	<b>basic</b>
Ties	<b>straddle</b>
Cell Height	<b>11 grids</b>



Option	Value
Xgrid	0.28
Ygrid	0.28
Routing Offset	xy-half

Save the file as `progen.db` under the `tech` directory.

### 2.3.3. Input Netlist

Following the instructions from Section 2.2.5.5, configure `ProSpin` to use the parameters shown in Section 2.3.3.1. Then use `ProSpin` to process the each of the cells in the netlist.

#### 2.3.3.1. Tutorial ProSpin Configuration

Option	Value
Output Directory	<code>cel</code>
HSpice Netlists	<code>net/all_netlists.net</code>
ProSpin Technology File	<code>tech/prospin.tcl</code>

### 2.3.4. Create Layout

Once the `.cel` files are created, follow the steps outlined in Section 2.2.6 to create the physical layout for the cells. Use the values from Section 2.3.4.1 from the command line, substituting the name of each cell where `cell` appears.

These values will create AGD and log files in the `agd` directory. View the layout as described in Section 2.2.7, using `progen.db` as the technology file.

#### 2.3.4.1. Tutorial ProGen Configuration

File	Path
<code>techfile</code>	<code>tech/progen.db</code>
<code>celfile</code>	<code>cel/cell.cel</code>

Chapter 2. ProGenesis Tutorial

<b>File</b>	<b>Path</b>
<i>agdf</i> ile	agd/cell.agd
<i>log</i> file	agd/cell.log