Physical Design using OpenLANE SKY130

Author

NIGIL MOHRA

1. Installating OpenLane v1.0

To install the necessary tools, please refer to the link below. Follow the instructions provided and run the example given to verify that the OpenLANE tools are installed correctly. Ensure that all commands are executed as the `root` user.

OpenROAD and some supporting tools.	OpenLANE Ubuntu Installation

During the initial `make smoke` test, some users might encounter the error:

```
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found
```

This error indicates a problem with loading the Qt platform plugin "xcb," which is necessary for opening the GUI.

This issue can be resolved by running the command `xhost +local:*` to allow local connections. This creates a workspace directory in your home directory if it doesn't already exist. This permits the root user on the local machine to connect to GUI apps. After completing the work and before closing the terminal, remember to revert the setting by running `xhost -local:*`.

2. Pilot Design

Before delving into the design flow. Installation of a text editor is required. The user can use any text editor. Here, `vim` is used. `Vim` can be installed by running the following command:

```
sudo apt-get install vim
```

2.1. Full Adder Design

All design files with the `.v` extension are stored in the `src` folder within a directory named after the design inside the `designs` folder of the OpenLANE directory. For instance, the one-bit full adder, implemented in Verilog, will have its files in this structure, and the GDSII file is generated using the OpenLANE ASIC flow.

Note: There are several methods to run the design using OpenLANE.

Create a folder named `fa` inside the `designs` folder using the following command:

```
sudo mkdir fa
```

To create the RTL file, either copy and paste the file directly into this folder or follow the alternative approach outlined below for a bit of extra fun!

Inside the `OpenLANE/designs/fa/src` folder, open the terminal and type the following command to create a file for the one-bit full adder:

```
cat > fa.v
```

After typing this command, enter the RTL code for the one-bit full adder. To finish and exit the terminal-based text editor, press `Ctrl + D`.

Next, open the RTL file in `Vim` with the following command:

```
vim fa.v
```

Add or modify the RTL design for the full adder as needed. To save the changes and exit `Vim`, use the command:

```
:wq
```

2.2. Configuration File

2.2.1. Creating the Configuration File

To run the one-bit full adder, a configuration file is required. Invoke the Docker container, inside the OpenLANE directory (which contains the `Makefile`), use the following command to invoke the Docker container:

```
sudo make mount
```

Once the Docker container is successfully invoked, Enter `tcl` interactive mode by running the following command:

```
./flow.tcl -interactive
```

To create a `config.tcl` file in the `designs` folder, use the script below. Replace `<design_name>` with the name of the main module, ensuring that it matches the module name and the file name of the RTL design:

```
./flow.tcl -design <design_name> -init_design_config -add_to_designs -config_file config.tcl
```

2.2.2. Modifying the Configuration File

To modify the contents of the `config.tcl` file, open the file in `Vim` as root user.

```
sudo vim config.tcl
```

Press `I` to enter `INSERT MODE` in `Vim`. Update the `config.tcl` file with the following content:

```
set ::env(DESIGN_NAME) {fa}
set ::env(VERILOG_FILES) [glob $::env(DESIGN_DIR)/src/*.v]
set ::env(FP_CORE_UTIL) 0.05
set ::env(FP_ASPECT_RATIO) 0.5
set ::env(RUN_CTS) 0
set ::env(DESIGN_IS_CORE) {1}

set tech_specific_config "$::env(DESIGN_DIR)/$::env(PDK)_$::env(STD_CELL_LIBRARY)_config.tcl"
if { [file exists $tech_specific_config] == 1 } {
       source $tech_specific_config
}
```

Press `Esc` to exit `INSERT MODE`. To save the changes and exit `Vim`, type:

```
:wq
```

If an error is encountered while saving, add an exclamation mark to force the save and exit:

```
:wq!
```

2.3. Running the Physical Design

To streamline the design flow and make it simpler, create the `fa_synth.tcl` file using the following command:

```
 cat > fa_synth.tcl
```

After entering the command, press `Ctrl + D` to exit. Open the `fa_synth.tcl` file in `Vim` and enter the following contents into the file:

```
package require openlane
prep -design fa -tag run1 -overwrite
run_synthesis
run_floorplan
run_placement
run_routing
run_magic
```

Save the changes and exit.

To run the PD, enter `tcl` interactive mode again using the previously described steps. Then, execute the following command to run the entire design flow:

```
source fa_synth.tcl
```

If all steps are followed correctly, the complete design flow is done.

3. Results

To inspect the `Floorplan`, `Routing`, `Placement`, and complete `GDS` file, the `klayout` tool can be used. Use the following command to install the `klayout` tool:

```
sudo apt-get install klayout
```

Launch `klayout`. The first step is to select the process technology `Sky130nm`. To set up the process technology, open terminal in the `OpenLANE` folder and switch to the root user by running:

```
sudo su
```

Use the following commands to navigate and download the `lyt` file. Use `ls` to list directories and ensure accurate navigation. Note that commands to navigate and list directories are not explicitly shown here; verify directory names as needed.

Command	Purpose
`cd`	Change directory.
`pwd`	Display the full pathname of the current directory.
`ls -a`	List all items, including hidden files, in the current directory.
`cd .volare`	Navigate to the `.volare` directory.
`cd sky130A`	Navigate to the `sky130A` directory.
`cd libs.tech`	Navigate to the `libs.tech` directory.
`cd klayout`	Navigate to the `klayout` directory.

Command	Purpose
`cd tech`	Navigate to the `tech` directory.

In the `tech` directory, three `.lyt` files will be present. Copy the `sky130A.lyt` file to the design folder.

```
cp sky130A.lyt /home/nigil/OpenLane/designs/fa/runs/
```

3.1. Technology Node Setup

To set up the technology node in `klayout`, follow these steps:

1. In the menu bar, click on `Tools` ➞ `Manage Technologies`.
2. In the dialog box that appears, add the `sky130A.lyt` file.
3. Press `OK` to confirm and exit the dialog.

In the workspace, locate the technology node setting. Change the technology node from `default` to `sky130nm`. This will configure `klayout` to use the `sky130nm` technology node for your design files.

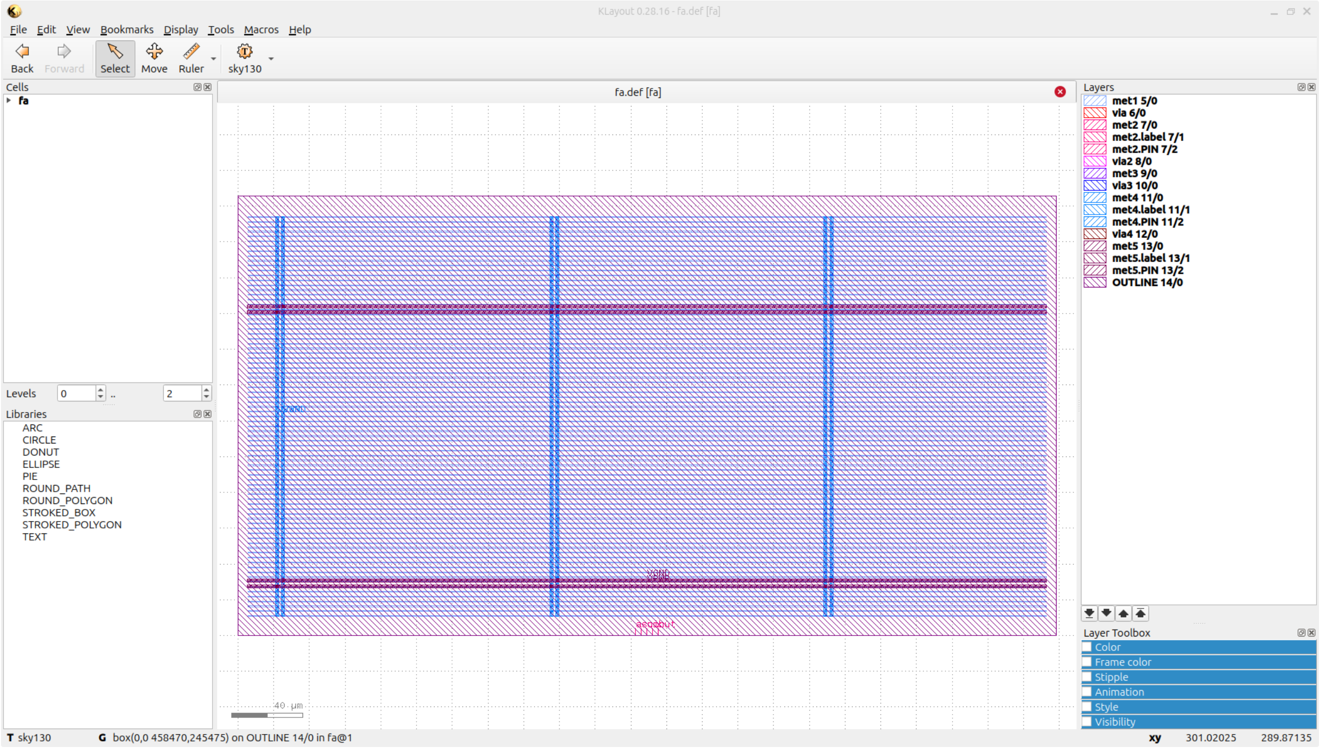
3.2. Checking the Layouts and Reports

To check the various layout results in `klayout`, add the Library Exchange File (LEF) by follwing these steps:

1. In the menu bar, click on `File` ➞ `Reader Options`. A dialog box will appear.
2. Navigate to the design folder, then to `runs` ➞ `tmp`, and find the file `<design_name_min.lef`.
3. Add this file to the dialog box under the `LEF/DEF` section.
4. Press `OK` to confirm and exit the dialog box.

Note: There may be multiple `.lef` files. Select the one that matches your process variation needs.

To view the results, go to `File` ➞ `Open` and navigate to the `results` folder inside the `runs` directory. Open any folder within `results` and select a `.def` file to view the layout in the workspace. Similarly, the `GDSII` file will be located inside the `signoff` directory. Navigate to this directory and open the `.gds` file to view it.



T sky130 G box(0,0 458470,245475) on OUTLINE 14/0 in fa@1 xy 301.02025 289.87135

Figure 1. KLayout Full Adder Floorplan

(Miscellaneous) Running with Constraints

To run a design with constraints, create a `.sdc` file. The `.sdc` (Synopsys Design Constraints) file, or `.xdc` (Xilinx Design Constraints) file, can be generated using Xilinx Vivado or an appropriate Synopsys tool. This file will specify the design constraints. Update the `config.tcl` File:

```
set ::env(BASE_SDC_FILE) $::env(DESIGN_DIR)/fa.sdc
set ::env(RUN_CTS) 1
set ::env(CLOCK_PERIOD) "100.000"
set ::env(CLOCK_PORT) "virtual_clk"
```

These settings will enable constraint processing, set the clock period, and specify the clock port. Follow the same procedures as before to generate the `GDSII` file. Running the design with constraints will produce detailed Area, Timing, and Power reports. The reports will be located in the `reports` folder inside the `runs` directory. Check this folder for the detailed reports on area, timing, and power.

References

1. OpenLANE : RTL-to-GDSII Flow - Part-1
2. OpenLANE : RTL-to-GDSII Flow - Part-2
3. Adding Your Designs - OpenLANE Documentation