

# OpenLane Tutorial - Hardening the Core

The chip core would usually have other macros inside it.

## Step 1: Starting the OpenLane environment

Just run the following commands to enter the OpenLane environment (from the installed location of OpenLane)

```
cd Openlane/  
make mount
```

## Step 2: Creating new designs

The following command creates a new configuration file for your design:

```
./flow.tcl -design <design name> -init_design_config -add_to_designs
```

This will create the following directory structure:

```
designs/<design_name>
```

```
|— config.json
```

```
|— src
```

## Step 3: Create the RTL files

You need to create or copy the RTL files (including black-box). The recommended location for files is

```
designs/<design name>/src/.v
```

## Step 4: Give the parameter configuration

You need to set the following environment variables in your configuration file for the chip core:

```
VERILOG_FILES
```

```
VERILOG_FILES_BLACKBOX
```

```
EXTRA_LEFS
```

```
EXTRA_LIBS
```

```
EXTRA_GDS_FILES
```

```
SYNTH_READ_BLACKBOX_LIB
```

```
MACRO_PLACEMENT_CFG
```

## **Step 5: Run the flow on the macro design**

Finally, run OpenLane. `flow.tcl` is the entry point for OpenLane. The command needs to be run from inside the environment of OpenLane as described in quick start.

```
./flow.tcl -design <design name> -tag full_guide -overwrite
```

## **Step 6: Analyzing the flow generated files**

You can open the interactive view using the following commands:

```
./flow.tcl -design <design name> -tag full_guide -interactive
```

```
package require openlane
```

```
run_synthesis
```

```
run_floorplan
```

```
run_placement
```

```
run_cts
```

```
run_routing
```

```
run_magic
```

```
run_magic_spice_export
```

```
run_magic_drc
```

```
run_lvs
```

```
run_antenna_check
```

```
or_gui
```

The above commands can also be written in a file and passed to `flow.tcl`:

```
./flow.tcl -interactive -file <file>
```

## **Step 7: Viewing of final layout**

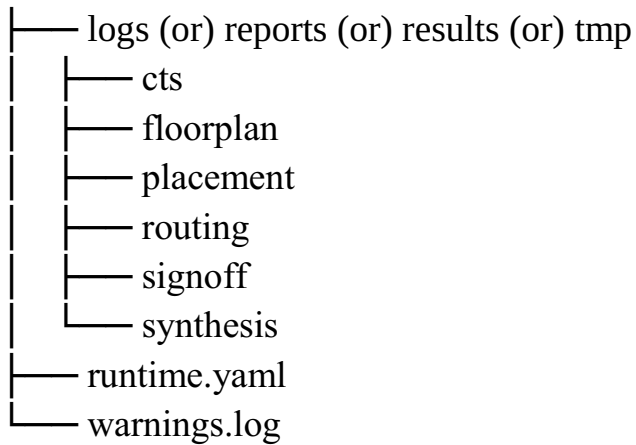
The following command generates a new gds or lef view for your design:

```
klayout designs/<design name>/runs/full_guide/results/final/gds or lef
```

(or)

```
magic designs/<design name>/runs/full_guide/results/final/gds or lef
```

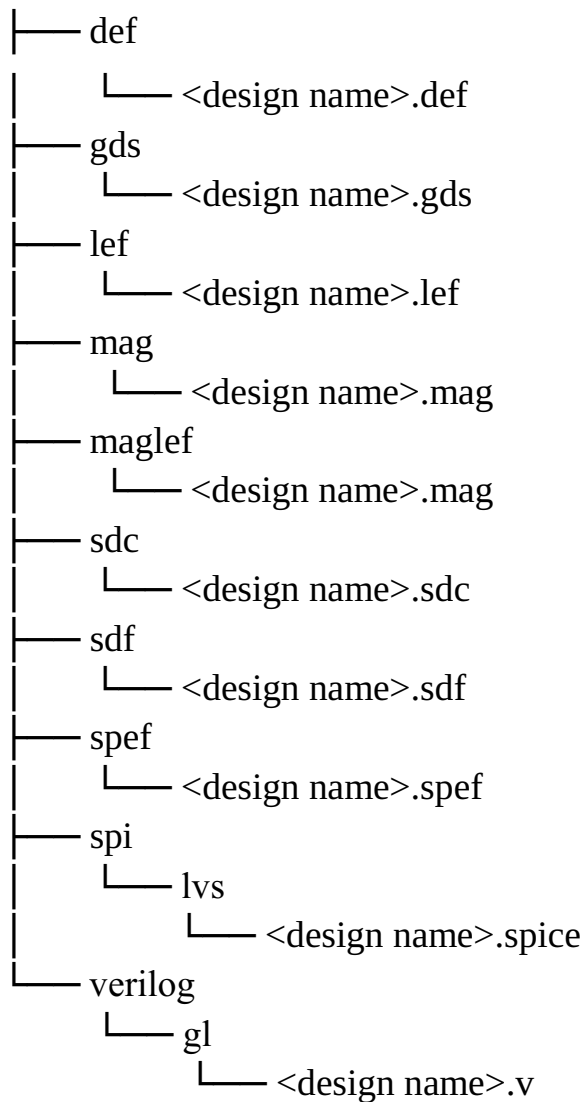
## Each run has following structure:



There are 4 directories logs reports results and tmp. In each of these directories, there are multiple directories. Directories are named according to the stage they belong to.

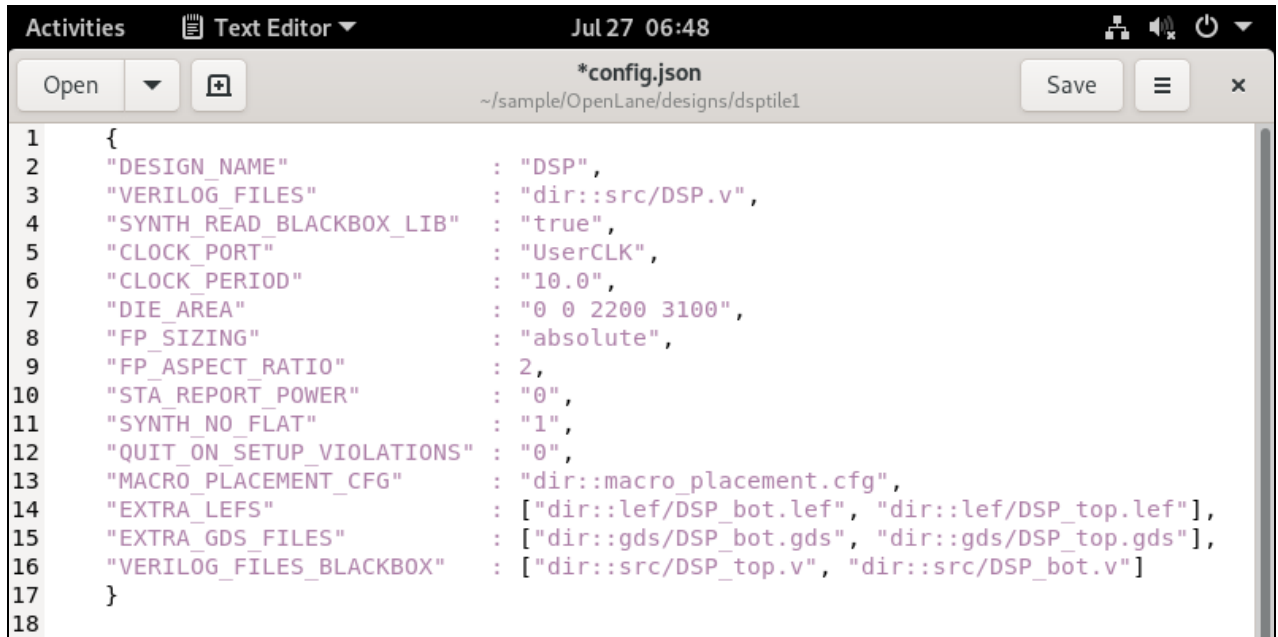
Finally output of OpenLane can be found in

[designs/<design name>/runs/full\\_guide/results/final](#)



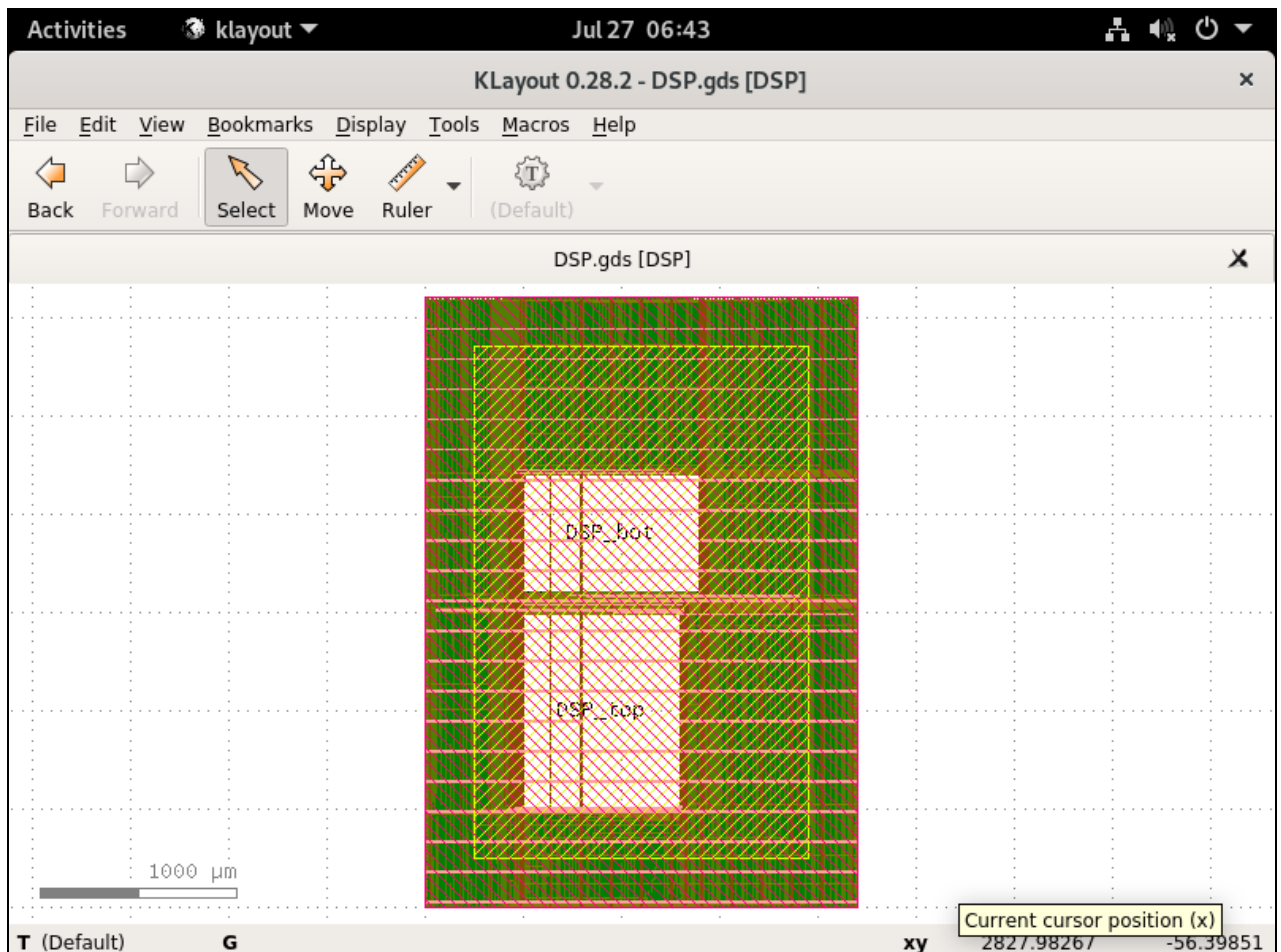
# Example - DSP Tile:

## Configuration Variables:



```
1  {
2    "DESIGN_NAME"           : "DSP",
3    "VERILOG_FILES"        : "dir::src/DSP.v",
4    "SYNTH_READ_BLACKBOX_LIB" : "true",
5    "CLOCK_PORT"           : "UserCLK",
6    "CLOCK_PERIOD"         : "10.0",
7    "DIE_AREA"             : "0 0 2200 3100",
8    "FP_SIZING"            : "absolute",
9    "FP_ASPECT_RATIO"      : 2,
10   "STA_REPORT_POWER"      : "0",
11   "SYNTH_NO_FLAT"        : "1",
12   "QUIT_ON_SETUP_VIOLATIONS" : "0",
13   "MACRO_PLACEMENT_CFG"  : "dir::macro_placement.cfg",
14   "EXTRA_LEFS"            : ["dir::lef/DSP_bot.lef", "dir::lef/DSP_top.lef"],
15   "EXTRA_GDS_FILES"       : ["dir::gds/DSP_bot.gds", "dir::gds/DSP_top.gds"],
16   "VERILOG_FILES_BLACKBOX" : ["dir::src/DSP_top.v", "dir::src/DSP_bot.v"]
17 }
18
```

## GDS View using KLayout:



## LEF View using KLayout:

