CENG 4120 Computer-Aided Design of Very Large Scale Integrated Circuits

Homework 3 (Part B)
Due Date: Apr 21, 2025 11:59pm

1 Setup Instruction

1. Setup and connect to the CSE VPN following the guideline here. OpenVPN is recommended.
2. For Windows users, MobaXterm is the suggested SSH client. Connect to server linux5-16 (e.g., linux10) using the following command,

```
$ ssh username@linux10.cse.cuhk.edu.hk
```

3. For Mac OS users, download and install XQuartz to enable showing the GUI for Innovus. Use Terminal provided by Mac OS to connect to server linux5-16 (e.g., linux10) using the following command,

```
$ ssh -X username@linux10.cse.cuhk.edu.hk
```

4. Upload **25hw3.zip** to the server (e.g., linux10). If you are using MobaXterm, just drag the file from local to remote. If you are using Terminal, open a new Terminal window (local) and upload the file using the following command,

```
$ scp -r /local/location/25hw3.zip
username@linux10.cse.cuhk.edu.hk:/remote/location/
```

Then, extract the files on the server and run the setup script.

```
$ unzip 25hw3.zip
$ cd 25hw3
$ source setup.sh
```

2 Place and Route Using Cadence Innovus

The two files **cla4_synth.v** and **cla4_synth.sdc** in the zip file are generated by following the instruction in homework1, with a clock period of 0.25.

We will start by creating a "multi-mode multi-corner" (MMMC) file named **mmmc.tcl**. It will configure the software to support multiple combinations of modes (a set of clocks, supply, voltages, timing constraints, and libraries) and corners (a set of libraries characterized for process, voltage, and temperature variations), and to evaluate them concurrently. This is to ensure that our design will work across a range of operating conditions. We only create the default mode and corner in this homework to make things easier. The **set_analysis_view** command specify the analysis view for both setup and hold time analysis. Don't forget to modify the path to **gscl45nm.lib** and make sure file **cla4_synth.sdc** is in the current directory. (In the mmmc file, you do not need to have the line break " \ ")

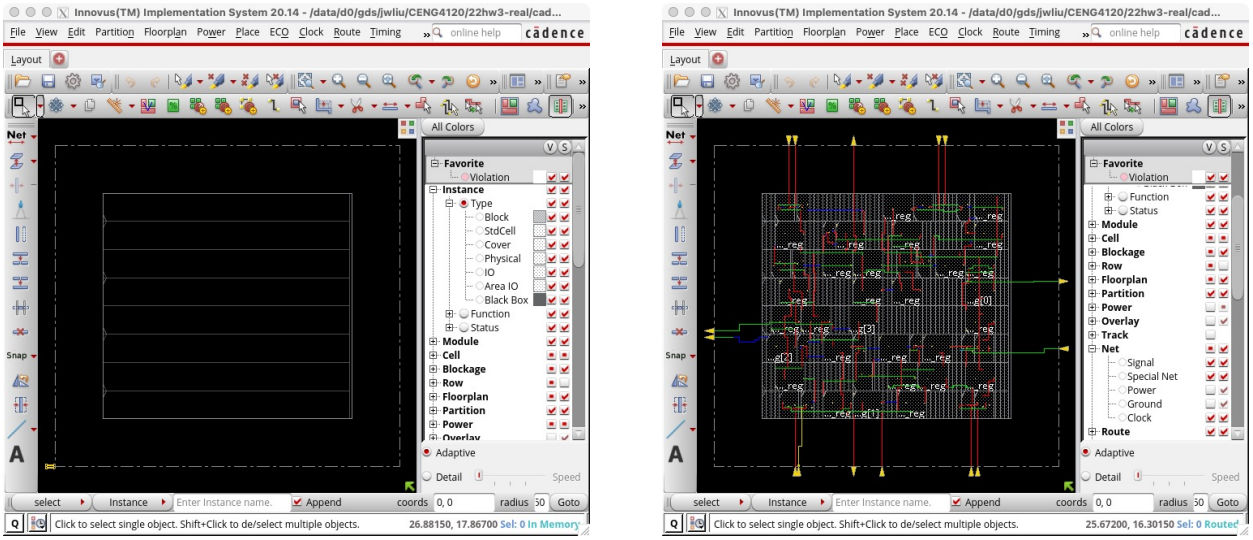
```
# mmmc.tcl

create_library_set -name libs_typical \
    -timing FreePDK45/gscl45nm.lib

create_constraint_mode -name constraints_default \
    -sdc_files cla4_synth.sdc

create_delay_corner -name delay_default \
    -library_set libs_typical

create_analysis_view -name analysis_default \
    -constraint_mode constraints_default -delay_corner delay_default

set_analysis_view -setup analysis_default -hold analysis_default
```

Start Cadence Innovus, you should be able to see the GUI popping up if you have followed the setup instruction.

```
$ innovus # start Cadence Innovus
```

We will first set some variables to tell Innovus the location of the gate-level netlist Verilog, the name of the top-level module in our design, the location of the **.lef** file, the names of the power and ground nets, and the location of the MMMC file.

```
innovus> set init_verilog cla4_synth.v
innovus> set init_top_cell cla4
innovus> set init_lef_file FreePDK45/gscl45nm.lef
innovus> set init_pwr_net vdd
innovus> set init_gnd_net gnd
innovus> set init_mmmc_file mmmc.tcl
```



(a) After Floorplanning (b) After Routing

Figure 1: Cadence Innovus GUI

After setting the necessary variables, we can initialize the design by

```
innovus> init_design
```

The next step is floorplanning. Since our design is very simple and has only one block, we only need to decide a few parameters for the floorplan. The six parameters are the aspect ratio (the ratio of the height divided by the width), row density (the percentage of the row area that will be occupied by useful standard cells), left, bottom, right and top margins respectively.

```
innovus> floorPlan -r 1.0 0.60 4.0 4.0 4.0 4.0
```

By inputting the above command, we set the aspect ratio to be 1.0, the row density to be 0.6 and all the margins to be 4.0 microns.

Now have a look at the GUI. You can use the menu option (Windows > Workspaces > Physical) to switch to the physical view, and use the option (View > Fit) to see the entire design. You should be able to see a square floorplan as in Figure 1a with rows that the standard cells will eventually be placed. In the following steps, you are encouraged to observe how the design will evolve after each command.

Next we will route the power and ground nets. We will first specify the names of the power and ground nets using the following command.

```
innovus> globalNetConnect vdd -type pgpin -pin vdd -inst *
innovus> globalNetConnect gnd -type pgpin -pin gnd -inst *
```

Construct power and ground rails on metal1 layer that will directly touch the standard cells.

```
innovus> sroute -nets {vdd gnd}
```

Create a power ring and some stripes to get power and ground to all standard cells.

```
innovus> addRing -center 1 -spacing 0.5 -width 0.5 \
    -layer {top 3 bottom 3 left 4 right 4} -nets { gnd vdd }

innovus> addStripe -nets {vdd gnd} -layer 4 -direction vertical \
    -width 0.4 -spacing 0.5 -set_to_set_distance 5 -start 0.5

innovus> addStripe -nets {vdd gnd} -layer 3 -direction horizontal \
    -width 0.4 -spacing 0.5 -set_to_set_distance 5 -start 0.5
```

Then, we can start to place our design by using the **place_design** command. This command will also generate a very rough routing solution to assist placement.

```
innovus> place_design
```

Add filler cells to fill in the empty spaces.

```
innovus> addFiller -cell FILL -prefix FILL -fillBoundary
```

Assign the IO pins' locations.

```
innovus> assignIoPins
```

It is time to do some serious routing.

```
innovus> routeDesign
```

After placement and routing, we need to check if there is any design rule violation and generate area, power and timing reports.

```
innovus> verify_drc
innovus> report_area
innovus> report_power
innovus> report_timing
```

Try zooming in and out in the GUI to have a close look at the design you have just finished. You can also show/hide some components using the panel on the right. Figure 1b shows the design after hiding the power/ground nets. Now we can exit Innovus.

```
innovus> exit
```

Similar to Design Compiler, we can also put all the commands we used in Innovus into a **.tcl** file and run all of them sequentially in one step.

```
$ innovus -files inv.tcl
```

Now answer the following questions,

1. Follow the above instructions to place and route. How many DRC violations are reported? What is the total area of the top module in square micron? What is the total power of the design in milliwatt? What is the slack in nanoseconds after placement and routing?

2. By using the **report_area** command, we can get the total area of the standard cells of each module, but we sometimes care more about the *die area*. The *die area* is basically the total area of the design, including the filler cells and empty areas. Smaller *die area* often allows us to fit more dies on a single silicon wafer and design more compact products. You can print the *die area* of your design using the following command. Note that the space between "designs" and ".area" is a must-have.

   ```
   puts "The die area is [get_db designs .area] square micron."
   ```

   Try different row densities (two decimal places) for floorplanning to minimize the die area after placement and routing. Do not change any other parameters. What is the smallest die area you can get without getting any DRC violations or errors?

3. How would total area and the slack in 1 change with row density? What might be the reason?

4. Attach a screenshot of the GUI showing the entire design you get in 2.