Microelectronic Systems

Lab 6

Physical Design: standard-cells based layout

May 30, 2023

Introduction

Here we are, ready to generate the layout of our designs and to check their **signal integrity** critical issues. The tool that you are going to use is **Innovus** by Cadence: the leader in the *physical design CAD market.*

As usual, copy the needed files from the following directory:

```
prompt> cp -r /home/repository/ms/cap6/* .
```

6.1 Physical design of the Ripple Carry Adder

Your first exercise will be working on a simple SUM architecture, organized as in Figure 6.1 and based on a **registered 128 bit RCA** *(the same you used in one of the previous labs)*. You are given the files that contain its description in VHDL, but also in Verilog *(sum.v)*, as Innovus only works with netlists defined through the latter file format.

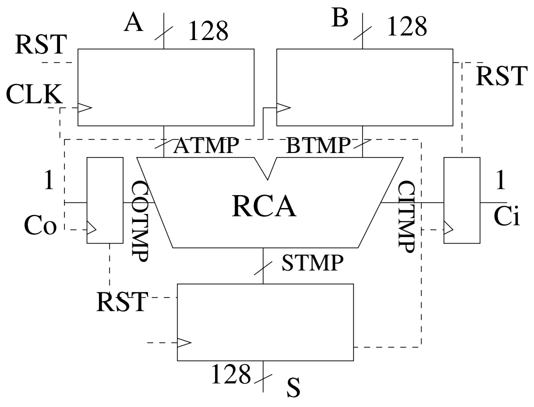


Figure 6.1: Structure of the Registered 128 bits RCA.

In the same directory, you also have a script file in which the synthesis constraints *(sum-t.scr)* are detailed. Please, read it carefully and notice the commands and the period considered. Finally, there are also the resulting timing and power reports obtained AFTER the synthesis *(sum_timeopt_t.rpt and sum_timeopt_pw.rpt)*. Again, please read them and notice the requested values of period and frequency. You will compare these quantities with those obtained after the layout phase.

You are now ready to start. Open a new shell and set the environment variables for Innovus by typing:

```
prompt> setinnovus
```

Then, launch Innovus:

```
prompt> innovus
```

⚠

Pay attention: this shell is now a dialog shell in which it is better if you do not write anything: just read the results every time you give a command inside the GUI.

Hints&Tips: In the following, you will find detailed information on how to proceed. However, remember that you can also read directly the Innovus manual, simply by typing: **help_cds_innovus** in the command line. Also the online help is useful, accessible by means of the top right box inside the GUI, especially if you want to learn more details about commands.

In order to process the physical design of the considered architecture several steps have to be taken into account:

Configuring → Floorplanning → Power planning and routing → Cell placing → Signal routing

The details of each of them are described below:

1) Configuring. The first step basically helps Innovus to find the path to the libraries used for the design and to read the synthesized description of the circuit.
From the top menu of Innovus' interface import the configuration file which sets the correct references to the libraries: **File** → **Import Design**. A new window opens: select **Load** from the menu below and then, from the browse menu inside the new window, choose the file **SUM.globals**. Notice the definition of the Verilog file *(.v)* containing the post synthesis cell view, the reference to the layout view of the library of cells *(.lef file)* and the definition of the power supply names, **vdd** and **gnd**. Notice also the *"Default.view"* file included. This last file stores information on the MMMC *(MultiMode MultiCorner)* analysis, i.e. references to files and data definitions organized and combined in order to consider not only default conditions, but also best case or worst case ones in terms of *temperature* and *process variations.* Now click **OK** in the "Design Import window": the design is imported with the correct cell reference and with the RTL Verilog description.
A **square** with several lines appears. This image represents a rough structure where several cells will be placed in the sequence of lines.

2) Structuring the Floorplan. At this step, Innovus sets the area to be assigned to the cell ensemble together with the area where the power supply ring will be routed, as shown in Figure 6.2.

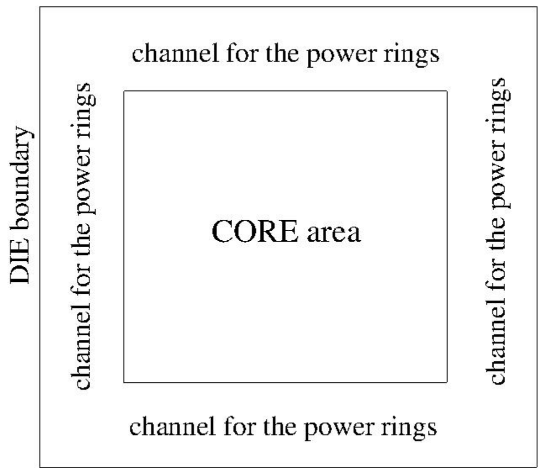

channel for the power rings
DIE boundary
channel for the power rings
CORE area
channel for the power rings
channel for the power rings

Figure 6.2: Area of the die dedicated to the core *(the place for the cells)* and to the power supply rings all around it.

From the top menu inside the Main window select: **Floorplan** → **Specify Floorplan**. An option window opens, allowing you to define the **Core aspect ratio** *(set it to 1.0)* and **Utilization** *(set it to 0.6)*. In section *"Core Margins by"* select **Core to die boundary** and force *5 (μm)* from the four sides of the core *(please note that the measurement unit is μm here)*. Click **OK** and see what happens. The cells height is already known at this point, as you can note by the grey horizontal lines. All the standard cells have the same height by construction. The width changes coherently with the transistors and interconnections area. At this point, Innovus knows how many rows will be needed for the design.

3) Inserting power Rings. The channel defined before will be filled with **two metal rings** for power and ground respectively. These metal stripes will be connected with the VDD and GND pads *(if this is the final chip, a wire-bonding package is supposed here; if not, more probable, these rings* *will be connected to other rings belonging to other blocks)* and will distribute hierarchically power and ground signals to the whole chip. For this reason, you will use for the rings a **high metal layer** and to avoid congestion, you will choose different layers for the definition of the horizontal lines *(M9)* and the vertical lines *(M10).*

From the top menu inside the Main window select: **Power** → **Power Planning** → **Add Rings**. An option window opens, allowing you to define:

- the **nets** to be associated to power and ground: click on the browse button, on the right end of the Net box, then select and add both *vdd* and *gnd*;
- the **type of ring**: in the *Ring Type* section leave the default configuration as *"Around core boundary"*;
- the **metal** associated to the rings: in the *Ring Configuration* section, select *M9 for top and bottom lines* and *M10 for left and right lines*; set all widths and spacing to 0.8, and select the option *Offset: Center in channel.*

Finally, click **OK**.

Now two rings have been designed: one for VDD and one for GND *(the connection to the electrical generator will be performed later).* Note that along the corners of the layout there are the **vias** for the connection between M9 and M10 metal layers. If you click on one of the present stripes, inside the bottom window you will receive information on the selected *metal layer*, on the *stripe geometry* and on its *coordinates.*

Question 1
What does this rectangle represent for the foundry that is going to produce your chip?

4) Inserting stripes. In this step **vertical metal wires** will be added, connecting the ring from top to bottom, and they will still be the VDD and GND lines. This process could be avoided, but it helps to correctly distribute the power and ground signals toward the chip center, so it is generally performed. The greater the number of the vertical stripes, the better the distribution. However, the congestion would be more problematic when also the other signals will be routed.

From the top menu inside the Main window select: **Power** → **Power Planning** → **Add Stripes**. A new window opens, where you have to:

- in the *Set Configuration* section, click on the browse button of the *Net box* and add both **vdd** and **gnd**; select *M10 for routing*, leaving the vertical option for the direction, and *set 0.8 for both width and spacing*;
- in the *Set Pattern* section, select **Set-to-set-distances** and choose a *value of 20*;
- in the *First/Last Stripe* section, select the **Relative from core or selected area** option and set the *Start* to 15, while leave *Stop* unconstrained.

Finally, click **OK**.

As you can see, **three sets of stripes** have been routed. Each set is composed of one stripe connected to GND *(click on it to see the properties)* and one stripe used as a VDD line. Use the ruler *(select it from the left menu)* to check the distances.

5) Standard Cell Power routing. This operation allows to **place horizontal wires** and, therefore, to prepare the VDD and GND wires for the standard cells. Such wires will be connected to the external ring and to the vertical stripes as well.

From the top menu inside the Main window select: **Route** → **Special Route**, then choose *vdd* and *gnd* as nets and click **OK**, leaving the default values.
Once the operation is concluded, you can observe the information about the *metal layer* and the *connectivity* by clicking the arrow in the left menu, sweeping on the metal stripes and reading the metal lines attributes in the bottom left window.

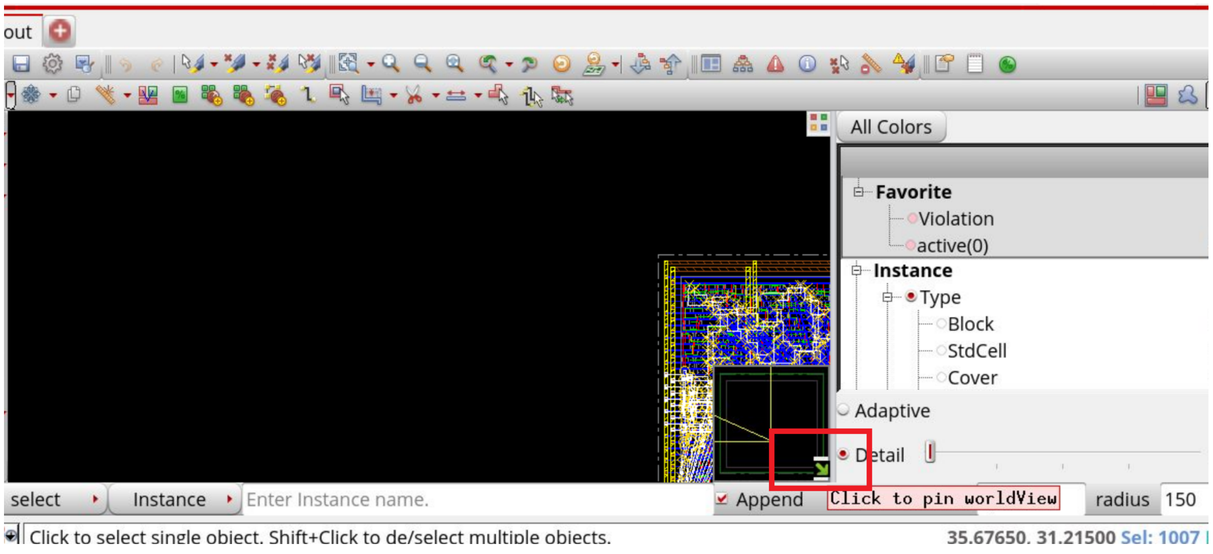


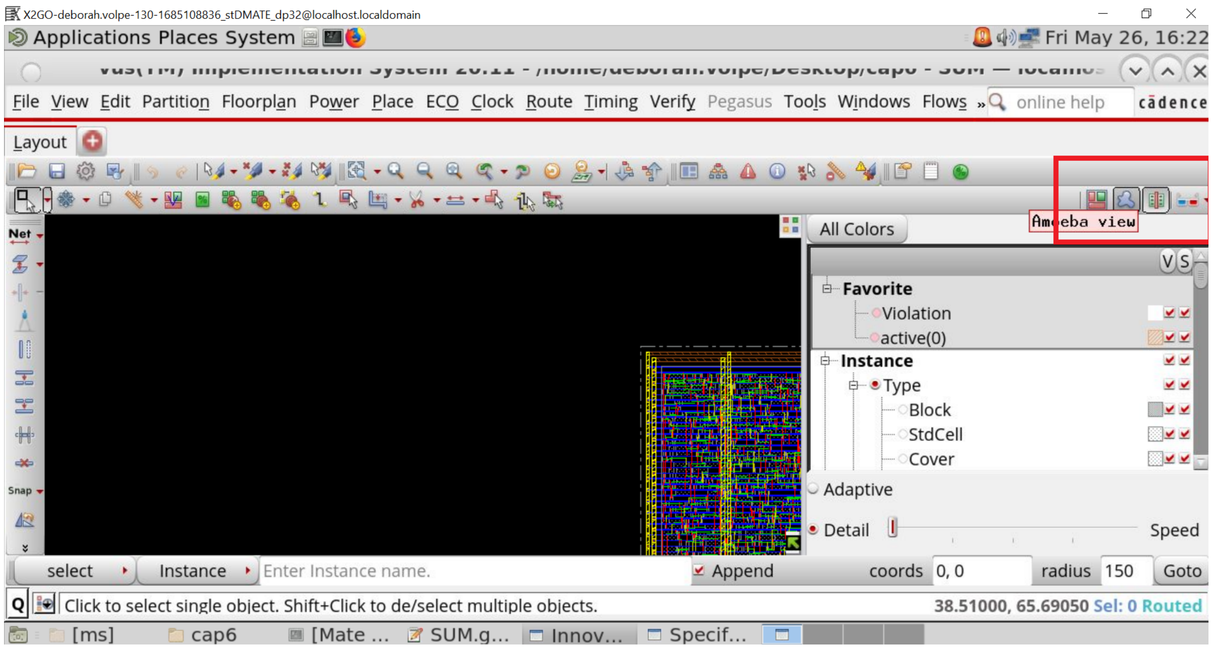
Figure 6.3: Arrow in the left for observing information on metal layer

6) Placement. Now the cells will be placed. Up to this point the only known thing was the **total area of the circuit** and the **number of rows** to be used. Hereinafter, the layout of each cell will have a unique position in one of the predefined rows.

Before starting the placement phase, select **Place** → **Specify** → **Placement Blockage** and then the layers from M1 to M8. In this way, the cells will be placed out from the space occupied by the power and ground stripes, to avoid congestion problems. Now click on **Place** → **Place Standard Cell** and finally confirm with **OK**. Observe what happens.

You can select the cells in order to read their name *(from the bottom left window)*, and zoom in the view for recognizing the input/output pin names. If you click on a pin you will see its connectivity *(virtual)* to the other pins. Along the die border you can observe the input and output pins that have be connected to the pads and, if you select one of them, its connection to the cell pins will be highlighted. What you see is only an abstract view of each cell, where some information is reported: its **size**, its **orientation** and both the **position and connectivity of its pins** which will be used during the routing phase for connecting the cells among them.

7) Exploring placed design From the second line menu, click on **Design Browser**. A new window useful to browse your circuit opens. Select, for example, **Modules** → **maprca**: all the gates included in the RCA block are highlighted. Choosing the "amoeba view" may be more helpful than the physical view. Now, play with the other sub-blocks.



Click to select single object. Shift+Click to de/select multiple objects.

Figure 6.4: Amoeba viewer

8) Placing I/O Pins From the main menu, click on **Edit** → **Pin Editor**. A new window opens allowing you to define along the edges of the DIE where to place the I/O pins. Inside the *Pin Group* section, where you can notice all the I/O signals defined in your top level module, select the first choice (**A**). Then, in the *Location* section select **Spread** and choose as *Spread Type*: **Along Side** if the signal is a *bus*, **From center** if the signal is a *single wire*. Finally, in the *Pin Attribute* section, force the *Side/Edge* option to **Top**.
Repeat the same steps for the other signals but changing the *Side/Edge* option. For example, you can place B on the left edge, S on the right and the other signals on the bottom.

Now, click **OK**. You will see all pins distributed along the edges of the DIE.

9) Post Clock-Tree-Synthesis (CTS) optimization. Before running the routing process, you can try to optimize our design to achieve the required timing constraints. In the Main window select **ECO** → **Optimize Design**. In the *Design Stage* section choose **Post-CTS** and in *Optimization Type* check both **Setup** and **Hold**. Then, click **OK**.

10) Place filler. It is helpful for technological reasons to complete the placement with **filler cells**. These elements will fill the free spaces in the layout to ensure continuity of the N+ and P+ wells in each rows. From the top menu select **Place** → **Physical Cell** → **Add Filler**. A new window opens where you have to click on **Select**, choose all the fill cells available and add them to the left list. These are different types of fill cells among which the placer will choose the appropriate ones for filling the placement gaps.

Now click **OK** and inspect what has happened.

11) Routing. This is the one of the last phases: the connection among the cells will be performed considering the available metal layers.

⚠

The already visible routing is only a logical connection among cells pins, that have been used during the placement for taking into account the interconnection length.

Click on **Route** → **NanoRoute** → **Route** and select **OK**.

Note that in the Innovus' shell a few messages are displayed on the routing iteration steps for this optimization, together with some interesting data on the performed routing. Among them, the **number of wires** used for each layer, the **total length routed** in each layer and the **number of via** considered. You can still analyze the metal layer used for the connections. For what concerns the *white boxes*, they represent violations which should be solved by a detailed operation that you will not perform.

12) Post routing optimization Notice that the design is at this point complete, even if we did not use real PADS all around the design. During this last step you will try to optimize the design to achieve the required **timing constraints**. Before starting this optimization operation, you have to issue on Innovus command line *(the shell from which you have launched Innovus)* the command **setAnalysisMode -analysisType onChipVariation**. Then, in the Main window, select **ECO** → **Optimize Design**. In the **Design Stage** section, choose **Post-Route** and in *Optimization type* check both **Setup** and **Hold**. Finally, click **OK**.

Before going on, save your routed view: **File** → **Save Design**. Check that as *Data Type*, **Innovus** is selected, and give the file a meaningful name. Hereinafter you will be able to import your design at the routed level again.

You can plot the obtained image of the circuit layout from the main menu: **Tools** → **Screen Capture** → **Screen Dump** and you can observe the results using **Tools** → **Screen Capture** → **Display Screen Dump**, where you have to select **All** as File Type.

We can now perform a few **analyses for timing** and **integrity**:

1) Parasitics. To analyze the behavior of the circuit over time, Innovus uses the values of the **parasitic resistance** and **capacitance** for each metal wire. During this step, such parasitics will be extracted. Innovus' engine is able to compute the resistance and capacitance associated to each rectangle using its attributes, technological and geometrical.

From the main menu select: **Timing** → **MMMC Browser**. Inspect the three possible **RC Corners** defined. We will use the standard ones.
Again from the main menu select: **Timing** → **ExtractRC** and choose all the possible files to save. Moreover, select **standard** as *RC Corner* and click **OK**.

Question 2

Once done, look at the file **SUM.spf** *(spf means Standard Parasitics Format)*: what does it represent? Note the *.subckt* instruction and the *"net section"* at the top of the file, and *"instance section"* at the bottom.
What about the **SUM.spef** file? Analyze the contained information and look for the net with the greatest capacitance and the greatest length.
Read the files **"SUM.setload"** and **"SUM.setres"**, which will be both used later: what do they force?

2) Delay. From the main menu select: **Timing** → **Report Timing**. A new window opens and as *Design Stage* choose **Post-Route**, while as *Analysis Type* choose **Setup**. Then, click **OK**. Repeat the same procedure choosing **Hold** in the *Analysis Type* section.
In the directory named *"timingReports"* you will find all the results produced by the timing analysis. The files with extension **.slk** and **.tarpt** contain both general and detailed information on the *timing paths* and eventual *violations*. The latter are referred to the timing constraints defined inside the

file **SUM.sdc**, which are identical to those that are considered during the synthesis, as they are already loaded with the initial configuration file.

⚠

Remember that **slack** means *"what is remaining between what you asked to have and what you actually have"*. So, if the slack is positive you are ok and you have spared some time, however if it is negative you are violating the constraint that you have imposed. Remember that if you don't have a real clock in the design, you have to specify a virtual clock to perform this step.

Now complete the timing analysis by selecting **Timing** → **Debug timing**. A new window appears, where an overview of the critical paths that have violated the constraint, i.e. the values respected at the synthesis level, are displayed. If you double-click on one of them, the first one for example, a new window appears which allows you to press element by element *(in the horizontal bar)* and to see the corresponding **delay**, putting into evidence the specific **path** inside the layout window. Play with these two windows in order to understand the potentials of their functionalities. It is also interesting to compare the delay found at the synthesis level with the one obtained at the physical design level.... play and LEARN!

3) Design analysis and verification. Before ending the place and route process you have to verify the **connectivity** and the **design rules**. For the first verification, select **Verify** → **Verify Connectivity** and then click **OK**. Check the message produced by Innovus and verify that there is **no violation**. Usually, violations are caused by *floating wires*.

You can now save the **area** and **gate count** data from **File** → **Report** → **Gate Count** and selecting **OK**. Finally, you have to save:

i. The post place and route Verilog netlist, from **File** → **Save** → **Netlist** and then confirming **OK**

ii. The file with delay annotation *(.sdf)*, from **Timing** → **Write SDF** and choosing **OK**.

💡 **Hints&Tips**: YOU ARE LUCKY or maybe not. No further analysis is required: due to problems with the new tool version and the new technology library, the **electromigration** and **IR drop** analyses will be skipped. You can, if you want, play looking at the manual at the section: *"Rail Analysis"*.

Summary of what is requested

A dump of the layout, a saved place and route file with the Innovus extension in the working directory, the extracted capacitance, resistance and ".spf" file, the Delay report.

6.2 Physical design of your ADDER

Here you can play with the ADDER you designed in the previous lab, following the steps detailed above to analyse how the critical path changes. To correctly execute the physical design you have to perform again the **synthesis** with the *correct wire model* and imposing the *correct constraints*. You can start from the provided script **sum-t.scr** and trying to modify it accordingly to your design. Don't forget also to modify the constraint for the CLK signal.

For the physical design you must create a new file **ADDER.globals** starting from the one named SUM.globals, considered up to now: make a copy of this last file, change its name to ADDER.globals and change the references to the Verilog file name. If the SDC file produced by the synthesis tool have a name different from *"SUM.sdc"* you should also update the **Default.view** accordingly. The references to all the libraries remains the same.

Summary of what is requested

A dump of the layout, the saved Innovus' design, the extracted capacitance, resistance and ".spf" file, the Delay report.

6.3 Physical design of your MULTIPLIER

Finally, you can play with the MULTIPLIER you designed in the previous lab, following the steps detailed above to analyse how the critical path changes. Of course, this will be useful for the DLX project.
The steps you have to consider are the same presented for the ADDER in Section 6.2.

Summary of what is requested

A dump of the layout, the saved Innovus' design, the extracted capacitance, resistance and ".spf" file, the Delay report.