UE VLSI-2 TP:
SoC Encounter

Matthieu Tuna

matthieu.tuna@gmail.com

0. Environment

```
> export LM_LICENSE_FILE=30000@licences.cnfm.fr
> export PATH=$PATH:/users/soft/opus/Linux/EDI-13.24.026/bin
> encounter
```

1. Design import

The first thing to do is to load the design into SoC Encounter. Click on the menu "File > Import Design". Use this form to import the netlists, the design libraries (timing: .lib and physical .lef) and the timing constraints:

a. The Netlists

In our case the verilog netlist is the netlist generated by RTLC. Click on the "..." button, then click on the button representing a directory. Find the netlist, click "Add" and "Close". For the top cell name, either you know the name of the top cell and you enter it ("By User") or you let the tool find it ("Auto Assign").

b. The Physical Libraries

For Technology/Physical Libraries, select LEF files and fill in the correct LEF files. What is a LEF file ?

c. Global Nets: Power Nets

For the Power section, enter the name of the power nets. For "Power Nets" enter, let's say "vdd_core" and for "Ground Nets" "gnd_core". Why do we have to enter this power net information ?

d. The Timing Libraries

Static Timing Analysis (STA) is done through different corner, this is called a multi-corner (MC) analysis. Latest technologies require lots of corners, however for our old technology an analysis with only two corners, Best Case & Worst Case, is enough.
Click "Create Analysis Configuration", then double-click "Library Sets". First we load the Worst case libraries. So enter for "Name: libsWC", then add all worst case .lib. Again, double-click "Library Sets" and load the Best case libraries. "Name: libsBC" and load the best case .lib.
You can see the two sets of libraries created under "Library Sets".

Now we create the different corner analysis. Double-click "Delay Corners". First "Name: cornerWC", and select "libsWC". Again, double-click "Delay Corners", "Name: cornerBC", and select "libsBC".

Now, we should load the timing constraints. We have only one sdc file. But in today's designs, different sdc can be provided to capture different modes of the design. For instance the functional .sdc (what we have) but you can have also an .sdc capturing the test (DfT) mode. This is called multi-mode analysis. This is why we have an MMMC (Multi-Mode Multi-Corner) analysis.
So, for timing constraints, double-click on "Constraint Modes", name it "const", and load your timing constraints.

Now we can create our different (two) corner analysis. Double-click “Analysis View”, Name: “setup”, Constraint Mode: “const”, Delay corner: “cornerWC”, Double-click “Analysis View”, Name: “hold”, Constraint Mode: “const” and Delay corner: “cornerBC”.

To specify which analysis views to use for Setup (max path) analysis, double-click on the Setup Analysis Views label in the left-side pane of the browser. Use the menu-pull down to select a view to add to the Setup view list.
To specify which analysis views to use for Hold(min path) analysis, double-click on the Hold Analysis Views label in the left-side pane of the browser. Use the menu-pull down to select a view to add to the Hold view list.

You can save the analysis view you have just created: “mips.view” and click “save&close”.

e. IO assignment file

Here we do not have any IO assignment file, leave it blank. What is an IO assignment file ?

Now you can save this configuration in a file to be used at encounter start in order to avoid re-entering those information again and again. To do so, just click on the “Save...” button and enter the name of the file such as “mips”. Click “Save”. Click “OK” and you can see in the main window the default created floorplan.

f. Check Design

As usual check if the design is completely loaded, run:

```
> checkDesign -netlist
```

Check the standard cell number, the area, the number of instances, the number of primary inputs/outputs. Is it ok ?

Are there multi-driver nets ? What is that ? Is it a problem ?
Now check that the SDC is correctly read, run:

```
> report_clocks
```

Is it ok ?

Explain L->L, L->T, T->L and T->T.

During the place-and-route phase, we will build the clock tree, therefore the clock should not be treated as “ideal” by the tool. Add the following SDC command at the end of your SDC file:

```
> set_propagated_clock [all_clocks]
```

Reload encounter with the new SDC and check with report_clock that the new command is correctly understood.

2. Floorplanning

To specify the floorplan click in the menu bar “Floorplan > Specify floorplan”. A new window should show up.

You can see beside the label “Core Utilization” the classical number 0.7. This specifies the core area, here it means that 70% of the design is filled with cells. You can play with this number to change the core area. This has an impact on the congestion of the design, in other words on the routability of the design. Explain why.

We will also add space between the core and IOs. To do so, in the core margins section, first check “Core to IO Boundary”, then, enter in the “Core to Left” 20, in the “Core to Right” 20, as well as in the “Core to Bottom” and “Core to Top”. Click “OK” and you should see the changes in the encounter window.

3. Power Planning

a. Connecting to Global Nets

During the design import step we have created two global power nets: vdd_core and gnd_core. During this step we will connect those global nets to the VDD and VSS pins of the standard cells. Click on the menu bar “Power > Connect Global Nets”. To connect the VDD pin of the standard cells to the global power vdd_core, click on “Pin”, and beside “Pin Name(s)” enter “VDD” which is the name of the power pin of all standard cells (you can check it in the LEF file). In the “Scope” part, click on “Apply All”. Then beside the label “To Global Net” enter “vdd_core”. Click on “Add to List”, you should see the command line appearing in the connection list window in the upper left corner. Do the same thing for “VSS” and “gnd_core”. Then click “Apply” and check if everything is correct ? No ? Please correct and rerun. Click “Close”. If you do not find the name of the power pins in the LEF try “vdd” and “gnd”.

b. Add Rings

In this step we will add a power ring around the core. Click on “Power > Power Planning > Add Rings”. In the “Net(s)” text label you should see our power nets “vdd_core gnd_core”. In the “Ring Configuration” part we can specify the metal layer we want to use for the core ring as well as the width and spacing between them. Change the “Top” and “Bottom” layers from M1 to M5, and “Left” and “Right” from M2 to M6. These layers are noted V, the first ones are noted H. Explain that. Change the width of all sides from 0.44 (which is the size indicated in the LEF file) to 1.32 (three times bigger). Change the spacing between the two nets from 0.46 to 0.92 . Regarding the “Offset” click on “Center in Channel” to get the core ring in the center of the channel between the core and the IOs boundary we have created earlier (during the floorplanning). Click “OK” you should see two rings added around the core. You can zoom/unzoom to see the rings and click on it to check wether the net is vdd_core or gnd_core.

c. Add Stripes

Now we will add stripes in our design. As usual, in the menu bar click on “Power > Power Planning > Add Stripes”. In the “Set Configuration” our power “vdd_core gnd_core” are printed besides the label “Net(s)”. For the “Layer” we choose “M6” which is vertical and for the “Width” the same value as for the rings i.e. 1.32, for the spacing 0.92 . For the “Set-to-Set Distance” we will put 50 in order to have a set of stripes every 30 microns. In the “Stripe Boundary” part we leave “Core ring”.

In the “First/Last Stripe” part we want to “Start From” “Left” and for the offset we put “X from left” to “25”. And click “OK”. You should see in the encounter window the stripes you have added. Zoom in to visually check that the stripes and the core ring are correctly connected with vias. By clicking on the those nets check that the vdd_core stripes is connected to the vdd_core ring and not to vss_core ring!

d. SRoute

In this step we will use the Special Router (SRoute) to route the “follow pins”. The follow pins are the horizontal stripes connecting all VDD and VSS metal chunks of the standard cells belonging to the same row. Choose “Route > Special Route”. As usual besides “Nets” you can see our “vdd_core gnd_core”. In our case we only want to connect the standard cells pins, therefore uncheck “Block Pins / Pad Pins / Pad Rings and Stripes”. We do not want “jog”, thus, uncheck “Allow-Jogging” and we do not allow SRoute to change layer (uncheck “Allow Layer Change”. Click “OK”, and on the encounter window you should see your follow pins: the horizontal blue stripes. Visually inspect that the follow pins are correctly connected to the vertical stripes as well as the core ring.

e. Saving the floorplan

Before saving the floorplan let's see if the created floorplan is legal. Use checkDesign to validate the floorplan:

```
> checkDesign -floorplan
```

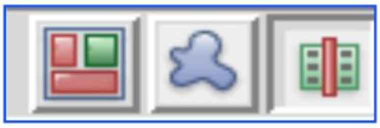
If everything is ok, you can save the flooplan in a file. Choose “Design > Save > Floorplan”. Then select only “Save Floorplan” and write in a 'fp' file. Then you just need to load this fp file to get your floorplan:

```
> loadFPlan <yourfile.fp>
```

4. Place Design

a. Standard Cells Placement

Choose in the menu bar “Place > Standard Cells”. We will run a full placement. Click “OK”, it should take only few seconds, and you should see in the encounter main window the placement of all the standard cells. If you do not see any standard cells, maybe you are in the floorplan view. Click on the physical view to see the placed standard cells:



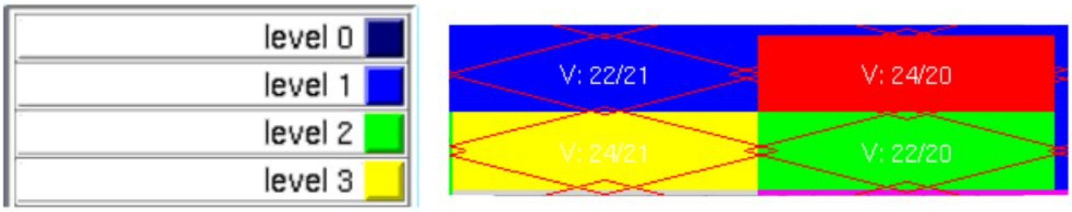
The command line to run the placement is “placeDesign”.

The placement done by default is timing-driven. If you want to specify another behavior or have more options, take a look at “Design > Mode Setup > Placement”. You can then choose to increase the congestion effort if your design is too congested for example. The command line for the placement mode option is : setPlaceMode -option1 -option2....

5. Trial Route

In order to get a quick estimation of the route feasibility and timing, the Trial Route will run a quick route without fixing DRC or LVS violations. Choose “Route > Trial Route” in the menu bar. Check “prototyping” and click “OK” and your design will be trial routed.

You can see the congestion of your design thanks to Trial Route. The congestion are represented by diamond symbols in specific colors such those in the figure below. If you do not see these diamond, you can, in the panel on the right, below “All Colors” uncheck “FPlan View”, this will uncheck the visibility of all element. When you have located the diamonds, you can recheck “FPlan View”. In the diamond symbol you can see a letter (which is either V or H for vertical congestion and horizontal congestion) and two numbers (e.g.: 24/20) which represent the demand / available resources. To overcome the congestion issue you might want to adjust block placements and/or orientations to make sure that connecting pins also face each other.



6. Timing Analysis

Running a timing analysis to check if the timing in the SDC file is met.

Choose “Timing > Report Timing”. Choose the correct “Design Stage” and explain your choice. This step can be done at every stage of the flow. In the “Analysis Type” part of the window check “Setup”. Click “OK” to run the timing analysis. In the command trace you should check if the the worst negative slack (WNS) is positive. Is the timing met ? Optimize the design with the optDesign command. Redo a timeDesign. Is the timing met now ? What is the slack ? Report the timing and check in the floorplan where is the path.

What is the TNS ? Why is important to have this information with the WNS ?

The command line is “timeDesign -preCTS”.
Let's optimize the design in order to get better results. Enter in the encounter shell:

```
> optDesign -preCTS
```

Then report the timing:

```
> report_timing
```

Is the timing better ? What is your WNS ? What is your TNS ?

7. CTS: Clock Tree Synthesis

The main priority for CTS is to minimize the clock skew. Choose “Clock > Synthesize Clock Tree”. CTS works with a specification file. To ease the process, you can click on the “Gen Spec” button in order to automatically generate a spec file.

At least we have to indicate which buffers and inverters CTS can use. Usually in the provided library there are some specific clock buffers. In our library you should see “CLOCKTREE”. Select it and the spec file will be written in the Clock.ctstch file.

Why are there specific cells for the clock tree ?

Is the timing met ? What is the WNS ? How many paths are violating the timing ?

To view the clock tree, in the menu bar, click “Edit > Find/Select Object...” in the “Criteria” part, select “Object Type” “Net”, in “Property” select “USE”, and in “Value” select “CLOCK”, finally click “Find” you will see the clock tree highlighted in the main window. If you zoom in, you can check that only “CLOCKTREE” buffers were used (tip: click on the “Q” in the bottom of the window to see the information of elements).

Is the timing met post CTS ?

What is your worst timing path ?

Maybe an optDesign postCTS will optimize that ? Is it better ?
Now that the clock tree is built, you can check if you meet the timing in hold mode. Run:

```
> timeDesign -postCTS -hold
```

Is the timing met in hold mode ?
Report the timing in hold mode with the report_timing command. In order to distinguish the setup and hold mode report_timing uses “early” and “late”. Which one is which one ? Explain.

8. Routing the design: NanoRoute

NanoRoute router is the routing solution of SoC Encounter. Choose “Route > NanoRoute > Route”.

We will run the global routing as well as the detail routing (the classical two steps of routing). To do so check “Global Route” and check “Detail Route”. You can click “OK”. As you can see NanoRoute is much longer than Trial Route. After NanoRoute, run a timing analysis:

```
> timeDesign -postRoute
```

Is the post-route timing met ? No ? Let's optimize the timing:

```
> optDesign -postRoute
```

Is the timing better ?
Please report the worst timing path. What is it ?

9. Adding filler cells

What is a filler cell ? Why do we need it ?
Choose “Place > Physical Cells > Add Filler”. Why do call these cells “physical cells” ?
Besides the label “Cell Name(s)” click “Select”. You can add all the cells and click “OK”.

10. Power Rail Analysis

In the menu bar choose “Power > Rail Analysis > Early Rail Analysis”.

Beside “Net Name(s)” choose “vdd_core” the “Voltage (V)” should be 1.08 and the “Limit (V)” should 10% lower, therefore: 0.972 . Before the analysis we have to provide a “Pad Location File” in order to instruct the tool where the power pads are placed in the layout. As we do not have such
file for the moment (neither placed power pads), we will create one. Click “Create”. In the new window create a pad for VDD whose coordinate are X = 0 and Y = 0, choose a “Layer”, M6 for instance, add this in the file, it will be our unique pad. Save the file and close the window. In the previous window puts this pad location file and click “OK”. On this new window click “Apply”. And you can see in the main encounter window the plot on your design. Normally the rails you have created should have green plots. If the visibility is not good enough, click on the “Advanced” tab and you have a seek bar to control the db transparency. An important feedback from this, is that you can check that all your power supply is correctly connected. To see the effect of the IR drop, in the “Auto Filter” part of the window, click on “Auto”, this will change the scale for the color. Click “Apply” and see in the encounter window the gradient of colors.

Therefore where is the best location for power pads ? Explain why ?

11. Timing Analysis

The goal is to meet the timing. To do so, remember the different steps and try to met the timing after each step:

- – preCTS:
  - – timeDesign -preCTS
  - – optDesign -preCTS
- – postCTS:
  - – timeDesign -postCTS
  - – optDesign -postCTS
  - – timeDesign -postCTS -hold
  - – optDesign -postCTS -hold
  - – (optDesign -incr)
- – postRoute:
  - – timeDesign -postRoute
  - – optDesign -postRoute
  - – optDesign -postRoute -hold
  - – (optDesign -incr)

For sure this training is not enough to tape out a chip, but provides a good start. Now dive into the reference manual!