Selected Topics in VLSI Design (Module 24513)

Course and contest – Intermediate meeting 3

Prof. Dirk Timmermann, Vlado Altmann, Jan Skodzik, Tim Wegner, Peter Danielis
Outline

1. Presentations of synthesized designs and results for ST65

2. Introduction of design flow
   - Final chip layout

3. Task and further infos for phase 4
Presentation of synthesized designs

1. Wiedenmann
2. Raddatz
3. Neubauer
4. Reinartz
5. Buescher
6. Lender

Consider: 5 minutes per presentation
Introduction to
Cadence Encounter Digital Implementation
Design-Flow

- Hardware Language
- Synthesis
- Layout

✓ Textual description of the circuit
✓ Transfer to logic gates
Layout–Flow I

Floorplanning

Basic structure of the chip (size, I/O, power supply …)

Placement

Placing the gates on the chip (core area)

Routing

Wiring of the gates, pads and power nets
Layout–Flow II

Floorplanning

Placement

Routing
Layout-Flow III

Floorplanning

Placement

Routing
Layout-Flow IV

Floorplanning → Placement → Routing
Layout–Flow V

Floorplanning

Placement

Routing
Layout-Flow VI

Floorplanning → Placement → Routing
Layout-Flow VII

Floorplanning → Placement → Routing
### Preparation for Encounter

1. Copy `encounter_14.tar` from `/home/tw346/share/` into your home directory (synopsys folder **MUST** be located in the same folder)
   ```
   <home>% cp /home/tw346/share/encounter_14.tar .
   ```

2. Rename Synopsys directory:
   ```
   <home>% mv synopsys_14 synopsys
   ```

3. Unpack the file and change to the encounter directory
   ```
   <home>% tar xfv encounter_14.tar
   <home>% cd encounter
   ```

4. Copy your final netlist into the netlist folder
   ```
   <home>/encounter% cp ../synopsys/results/your_filter.v ./import/netlist/
   ```

5. Execute `start_encounter.sh` in the encounter folder to start Cadence EDI
   ```
   <home>/encounter% ./start_encounter.sh
   ```

**Remark:** The names of the nets in your final netlist may not include slashes, backslashes or squared brackets
Overview

- Menu
- View options
- Tools
- Visibility / Selectability
- Global View
Design-Import I

1. File → Import Design
2. Loading a configuration file according to your chosen supply voltage (Sets the paths of the timing libraries and LEF files)
   - VLSI_project_10V.conf or
   - VLSI_project_11V.conf or
   - VLSI_project_12V.conf or
   - VLSI_project_13V.conf
3. Include the Verilog files:
   - import/floorplan/CORE65LPSVT_floorplan.v
   - import/floorplan/CORE65LPHVT_floorplan.v
   - import/floorplan/CORE65LPLVT_floorplan.v
   - import/netlist/your_filter.v
4. Top Cell: your_filter
5. Change to the Advanced tab

6. Select “Power“

7. Assign global nets for power:
   - Power nets: VDD
   - Ground nets: GND
8. Set the capacitance files for “RC extraction“:

Best Capacitance Table File: /opt/lib/cmos065/adv_EncounterTechnoKit_cmos065_7m4x0y2z_4.2/TECH/cmos065_7m4x0y2z_Best.captable

Worst Capacitance Table File: /opt/lib/cmos065/adv_EncounterTechnoKit_cmos065_7m4x0y2z_4.2/TECH/cmos065_7m4x0y2z_Worst.captable

Hint: If /opt/lib/ does not appear in the path browser, type /opt/lib/ in the load menu to mount the directory.
9. Save the import settings in your encounter directory: `<home>/encounter/<name>.conf` (Allows to reload default settings)

10. Click OK to complete import. First layout appears

Hint: It is advisable to save the complete design after each layout step (File → Save Design → Encounter).
Floorplanning

General steps for floorplanning:

• Determination of chip and core size
• Placement of pads
• Creation of the rows for gate placement
• Placement of macro blocks (memory, PLL, DAC, ...)
• Applying power supply
Floorplanning I

1. Floorplan → Specify Floorplan
   (1) Modify Core Utilization and Core Margins to suit your needs

2. Click OK
   Modified layout appears
Floorplanning II

3. Power \(\rightarrow\) Connect Global Nets...
   (Connect all power pins to the global power nets)
   (1) Instance Basename: *
   (2) Pin Name(s): vdd
   (3) To Global Net: VDD
   (4) Add to List
   (5) Instance Basename: *
   (6) Pin Name(s): gnd
   (7) To Global Net: GND
   (8) Add to List

4. Apply and Cancel to close
Floorplanning III

5. Power → Power Planning → Add Ring…
   (1) Net(s): GND VDD
   (2) Ring configuration: 2 µm
      (can be adapted to your needs)

6. Click OK
   Modified layout appears
**Placement**

Placement of standard cells. Goals:

- Connections as short as possible
- Wireable design

1. Place → Place Standard Cell...
   
   Deselect “Include Pre-Place Optimization”

2. Click OK

Modified layout appears after change to physical view:
Routing / Floorplanning

1. Edit → Pin Editor…
   (1) Pin Group: x_in[ ]
   (2) Side: Left
   (3) Spread: Along Entire Edge
   (4) Apply
   (5) Pin Group: y_out[ ]
   (6) Side: Right
   (7) Spread: Along Entire Edge

2. Click OK
Routing

General remarks on routing:

• Wiring of power nets
• Wiring of signal wires between the standard cells and pads
• Up to 7 metal layers
• Nano Route:
  - Wiring of power nets for complete power supply
  - Global Routing
  - Detailed Routing
  - Timing driven possible
Routing I

Routing of power nets

1. Route → Special Route
   Deselect “Block Pins”, “Stripes"

2. Click OK
   Modified layout appears
Routing II

Routing of cells

3. Route → NanoRoute → Route...

4. Click OK
   Modified layout appears
Timing Analysis I

1. Options  →  Set Mode  →  Specify Operating Condition/PVT

Tab max (Setup Time Analysis)
- Worst case condition necessary: high temperature, bad process, low voltage

For better comparison to former results we apply nominal conditions
- Nominal condition: typical temperature, process and specified Vdd (1 – 1.3V)
Timing Analysis II

Tab min (Hold Time Analysis)
  – Best case condition: *low temperature, best process, high voltage*

2. Click OK
Timing Analysis III

3. Options → Set Mode → Specify Delay Calculation Mode
   Select “SignalStorm“

4. Click OK
Timing Analysis IV

Timing constraints necessary
Most important:
- Clock declaration
- Exception for the reset

5. Timing → Load Timing Constraint
Select file: timing.constr

6. Click OK
Timing Analysis V

7. Timing → Extract RC
   (1) Select „Save Cap to“
   (2) Select „Save Setload to“
   (3) Select „Save Set Resistance to“
   (4) Select „Save SPEF to“

8. Click OK

9. Timing → Write SDF

10. Click OK
Timing Analysis VI

11. Timing → Report Timing
   (1) Select „Post-Route“
   (2) Select „Hold“ time analysis
   (3) Check, if a hold time violation occurs (see next slide)

12. Click OK

   (1) Select „Post-Route“
   (2) Select „Setup“ time analysis
   (3) Calculate your clock period

14. Click OK

Information for the slack is given at the terminal (see next slide):
- Negative slack: Violation
- Positive slack: No violation
Timing Analysis VII

Based on setup analysis:  \( T_{\text{min}} = \text{clk} - \text{slack} \)

Ex.: `create_clock -name "clk" -period 1.5 -waveform {0.0 0.75} {clk}

\( \text{clk} = 1.5 \text{ ns} \)

\[ T_{\text{min}} = 1.5 - (-0.034) = 1.534 \text{ ns} \]
\( f_{\text{max}} = 651.9 \text{ MHz} \)

\[ T_{\text{min}} = 1.5 - 0.166 = 1.334 \text{ ns} \]
\( f_{\text{max}} = 749.6 \text{ MHz} \)
Timing Analysis VIII

Further specific timing information is available:
- Clock declaration
- Exception for the reset
- Path slacks
- Number of violated paths
- Path highlighting
- ...

15. Timing → Debug Timing
16. Click OK
Power Analysis

1. Change to the synopsys folder
   
   `<home>/encounter% cd ../synopsys/`

2. Adapt frequency and file name of the according verilog file in the "filter_power_cadence.tcl" script
   
   • Default path and file name:  `./results/your_filter.v`
   • If you changed the netlist with Encounter you need to update this file!
   • Set achieved frequency after layout and your operation conditions
   • For correct power values: frequency needs to be adapted in the test bench

3. Start the power script, this will convert the parasitics file from Encounter and start Synopsys Design Vision
   
   `<home>/synopsys% ./start_power_analysis.sh`

4. In Design Vision
   
   `File → Execute Script → filter_power_cadence.tcl`

5. Power values in the final power report: “./results/design_report.txt”
Tasks and further infos for phase 4
Tasks for phase 4

- Target: Presentation of results for a working and optimized design layout (Observe/discuss differences to Synthesis results)
- Target Benchmark:

\[ \text{Metric} = \frac{f^4}{P_{\text{leak}} + P_{\text{dyn}}} \]

- \( P_{\text{leak}} \) and \( P_{\text{dyn}} \): "/results/cadence_power_report.txt"
- F: “filter_power_cadence.tcl”

- Consider impact for your design:
  - Differences to synthesis results
  - Layout targets / constraints
Handover infos for phase 4

• Final firm deadline: December 10th 2014, 23:59 CET
• Email your presentation (ppt, pptx or pdf)

• Design Handover:
  1. Copy your latest verilog netlist (of your final/best chip layout):
     cp /home/<your_login>/encounter/import/netlist/your_filter.v
     /home/tw346/share_in/projekt2014/<your_login>/
  2. Save your final chip layout in Cadence Encounter in your Cadence
directory: File ➔ Save Design ➔ as type Encounter
     File name: “your_filter_<your_login>.enc” (e.g. your_filter_at101.enc)
Handover infos for phase 4

3. Please verify that your saved design and your parameters are correct:
   (1) Restart Cadence Encounter
   (2) **Design → Restore Design** (choose Design from step 2: your_filter_<your_login>.enc)
   (3) Rerun the timing AND power analysis, record the final achieved values

4. Copy your final layout and its directory:
   ```
   cp /home/<your_login>/encounter/ your_filter_<your_login>.enc
   /home/tw346/share_in/projekt2014/<your_login>/
   (e.g. cp /home/at101/encounter/your_filter_at101.enc /home/tw346/share_in/projekt2014/at101/)
   
   cp –r /home/<your_login>/encounter/ your_filter_<your_login>.enc.dat
   /home/tw346/share_in/projekt2014/<your_login>/
   (e.g. cp –r /home/at101/encounter/your_filter_at101.enc.dat /home/tw346/share_in/projekt2014/at101/)
   ```
Handover infos for phase 4

5. Check the transfer directory, should be similar like this:

/home/tw346/share_in/projekt_2014/<your_login>/

    your_filter.v     (file)
    your_filter_<your_login>.enc  (file)
    your_filter_<your_login>.enc .dat  (dir)

6. Email the following results to vlado.altmann@ uni-rostock.de

/cc tim.wegner@uni-rostock.de:

(1) Frequency, period
(2) Power (dynamic and leakage)
(3) Number of pipeline stages
(4) Benchmark metric for the ASIC
(5) Core size, core utilization
Mandatory infos for results display

Include these elements in your presentation:

- Picture of chip layout
- Frequency response
- Path histogram
- Design parameters (with correct units)

<table>
<thead>
<tr>
<th>Timing ( (T_{\text{min}} / f_{\text{max}}) )</th>
<th>743 ps / 1345.9 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power ( (P_{\text{dyn}} / P_{\text{leak}}) )</td>
<td>106.83 mW / 2 µW</td>
</tr>
<tr>
<td>Pipeline Stages</td>
<td>2</td>
</tr>
<tr>
<td>Benchmark</td>
<td>[x]</td>
</tr>
<tr>
<td>Core size</td>
<td>90777 µm²</td>
</tr>
<tr>
<td>Core utilization</td>
<td>90.09 %</td>
</tr>
</tbody>
</table>
Pad Insertion I

• For a real chip design, pads are missing
• Irrelevant for the contest because of their immense impact on power consumption and delay

• To create a chip with pads, the following has to be done:
  – Requirements:
    • all files and folders have to be in the folder structure presented in the tutorials

1. Change to the encounter folder
   `<home>% cd encounter/

2. Start the encounter script with pad insertion:
   `<home>/encounter% ./start_encounter_pads.sh
   • in the following a netlist with pads will be generated by Synopsys Design Compiler and the verilog file will be copied to the /import/netlist/ folder
Pad Insertion II

3. Encounter will be started automatically
4. Load the \textit{VLSI\_project\_pads.conf} in the Design $\rightarrow$ Design Import section (all required files will be included)
5. Ok
6. Layout your chip as explained before (start on slide 20)
7. Three exceptions:
   • Do not execute the Pin Editor!
   • For connecting global nets: Two additional triples are necessary
     
     \begin{tabular}{ll}
     Instance Basename: & UPC \quad UGC  \\
     Pin Name(s): & VDD \quad GND  \\
     To Global Net: & VDD \quad GND  \\
    \end{tabular}
   
   • Power analysis (in the synopsys folder):
     
     – Change the frequency in the \texttt{filter\_power\_cadence\_pads.tcl} -file
     – Start with \texttt{/start\_power\_analysis\_pads.sh}
     – Execute: \texttt{filter\_power\_cadence\_pads.tcl}
Remarks for work and presentation

Possible fields for investigation (all have to include layout phase):

• Core row utilization
• Core and chip size
• Core position
• Aspect ratio of core area
• Pin positions
• Placing options (Global, incremental, timing driven …)
• Routing options (NanoRoute, WRoute, Timing driven, DFM …)
• Size and spacing of power rings
• Different power nets (rings, grids, stripes …)
• Limited metal usage
• Different gate libraries
• Application conditions
• Clock tree synthesis
• Manual placement and routing
• Architectural impact on ASIC and FPGA
• DFM: Design For Manufacturing (Via doubling, Via minimization, Metal fill, Wire spreading …)
• Electromigration
• Antenna fixing
• …
Questions?

Next meeting: December, 11th

Target: Presentation of complete chip layout and its results. (Observe differences of the design phases)