

ECE1388 Project 3

Digital Circuit Synthesis and Place-and-Route: Unsigned 4x4 Array Multiplier

In this project, you will design a digital multiplier using Verilog, synthesize the design using standard cells from the TSMC 65nm technology, and place & route your synthesized netlist to obtain a Cadence layout of the digital circuit. Section 11.9.1 in the textbook will provide you with a starting point on unsigned array multiplication, as well as an example of a 4x4 array multiplier.

You have been provided with the following files to complete Project 3:

- Verilog test bench for functional verification of your unsigned 4x4-bit multiplier Verilog
- Synthesis script to synthesize your Verilog multiplier using Synopsys Design Compiler
- Cadence Innovus configuration file to set up the Cadence place & route environment

You will also need to copy & paste the TSMC 65nm Standard Cell Verilog file from the following network location to your project working directory:

```
/CMC/kits/tsmc_65nm_libs/tc65gplus/TSMCHOME/digital/Front_End/verilog/tc65gplus_140b/tc65gplus.v
```

This file contains the Verilog descriptions of all the standard cells in this technology and must be included to perform functional simulation of the synthesized and placed & routed designs.

Include the following lines in your .cshrc file:

```
source /CMC/tools/CSHRCs/Cadence
source /CMC/tools/CSHRCs/Cadence.SOC
source /CMC/tools/CSHRCs/MMSIM.72
source /CMC/tools/CSHRCs/Cadence.EXT
source /CMC/tools/CSHRCs/Synopsys.2017.09
source /CMC/tools/CSHRCs/Cadence.INNOVUS.18
source /CMC/tools/CSHRCs/Synopsys.Formality
source /CMC/tools/CSHRCs/Cadence.INCISIVE.15
```

Part 1: Verilog design of an unsigned 4x4 array multiplier

1. Using Verilog code, describe the digital logic of an unsigned array multiplier to calculate the unsigned product of two unsigned 4-bit numbers. Provide the full Verilog code in your report.
2. You have been provided with an exhaustive Verilog test bench to verify the functionality of your 4x4 multiplier. You can use NCVerilog to run the test bench with the following Linux command:
`ncverilog multiplier.v testbench.v`
Note that you must include ``timescale 1ns/1ps` before your multiplier module declaration.
You may also use Modelsim if you prefer, but NCVerilog is more straightforward.
Provide the output of the test bench dialogue in your report to show the functionality of your multiplier.

Part 2: Synthesis using standard cells

1. Place the `synthesis.tcl` file in the same working directory as the Verilog test bench and your multiplier design. This TCL script will be launched using the Synopsys Design Compiler tool to synthesize your multiplier using the digital standard cells in the TSMC 65nm library. Run the Design Compiler GUI with the following Linux command: `design_vision`
Perform the synthesis of your Verilog multiplier design by executing the synthesis script in Design Vision: File > Execute Script
The compilation (`compile_ultra`) will take a few minutes to complete. Once complete, you will be able to view the gate-level schematic of your synthesized multiplier by right-clicking on the multiplier design in the Logical Hierarchy window and selecting the schematic view.
Save a copy of the schematic: File > Print > Print to File (PDF).
2. The Synopsys Design Compiler tool will output the following files into the working directory:

- Synthesized Verilog netlist: multiplier_syn.v
- Area constraint file: multiplier_syn.sdc
- Timing and Area reports

In your report, provide the generated estimates for total area and critical path timing. If you were to clock the design, what would be the maximum clock frequency? In your report, include the schematic of the synthesized multiplier, and highlight the critical path based on the timing report.

3. Verify that your synthesis was successful by running the multiplier test bench with the newly-synthesized netlist and the TSMC 65nm standard cell Verilog technology file:
ncverilog testbench.v multiplier_syn.v tcbn65gplus.v
Prior to running NCVerilog, ensure that all the module names in the generated Verilog files match.
Provide the output of the NCVerilog test bench dialogue in your report.

Part 3: Place & route

For this design, you will only be placing & routing the multiplier IP block, without a pad frame. Once fully placed & routed, the multiplier block can be exported from Cadence INNOVUS to Cadence Virtuoso layout editor for integration with other layout blocks. (This is not part of the assignment)

1. Place the innovus.globals and multiplier.view file in the same working directory as the multiplier_syn.v and multiplier_syn.sdc files. Run the Cadence INNOVUS place & route tool with the following command: innovus.
Set up the design: File > Import Design > Load > innovus.globals
This will load the appropriate timing and physical cell libraries for the place & route.
2. Define your design's floorplan: Floorplan > Specify Floorplan.
Ratio (H/W): 1, Core utilization: 0.7.
Press Apply and OK.
3. Place IO pins using the INNOVUS Pin Editor: Edit > Pin Editor
Place operand A: *Pin Group: A[], Spread: From Starting Point, Side/Edge: Left, Layer: M3, Offset based: 1, Spacing: 3, Start to end Direction: Clockwise.*
Place operand B: *Pin Group: B[], Spread: From Starting Point, Side/Edge: Right, Layer: M3, Offset based: 1, Spacing: 3, Start to end Direction: Counterclockwise.*
Place output product: *Pin Group: product[], Side/Edge: Top, Layer: M2, Spread: Along Entire Edge, Offset based: 0 0.*
4. Place power stripes for VDD and VSS: Power > Power Planning > Add Stripe.
Add VDD VSS to Net(s) .
In Basic, Select the M4 Layer with *Width: 1, Spacing: 1 and Set-to-set distance: 4. Press Apply*
Select the M5 Layer with *Width: 1, Spacing: 1 and Set-to-set distance: 4. Press Apply and OK.*
You have now set up the power connections for the VDD and VSS rails.
5. Place the standard cells: Place > Place Standard Cell > OK
Once complete, you can view the placed cells: Place > Display > Spare Cell
Check the design for geometry errors: Verify > Verify Geometry > OK
Note: At this stage in the design, the placement tool will perform coarse-grained routing and DRC violations may be ignored.
6. Route the power connections: Route > Special Route. In Special Route, click on Via Generation tab, Via Generation > Make Via Connections To: *check "Stripe". Press OK.* You should now see the VDD and VSS rails of the standard cells connected to the VDD and VSS stripes that you placed.
7. Route the signal connections: Route > NanoRoute > Route > OK
Verify that the design does not have any geometry errors: Verify > Verify Geometry > OK

8. Add filler cells to fill the unused area of your design: Place > Physical Cell > Add Filler
Under Cell Name(s), select all the available filler cells from the cell list (FILL1, FILL2, FILL4, FILL8, FILL16, FILL32, FILL64) by clicking on Select.
Verify that the design does not have any geometry errors: Verify > Verify Geometry > OK
9. Set up a power analysis: Power > Power Analysis > Setup. *Analysis View: setup_view*.
Run the power analysis: Power > Power Analysis > Run
Perform the power analysis for a constant input activity factor of 0.2, over the frequency range of 100MHz to 1GHz. Plot your results for total internal power and total switching power vs. frequency on the same graph.
10. Obtain the total area of your placed & routed design: File > Report > Summary
Compare the resulting total core area after place & route with the estimated synthesis area from Part 2.2. Comment on the discrepancy.
11. Include in your report a screen capture of your placed & routed design in Cadence INNOVUS.
12. Export the Verilog netlist of the placed & route design: File > Save > Netlist
Save the file as multiplier_netlist.v, and verify that your synthesis was successful by running the multiplier test bench with the newly-synthesized netlist and the TSMC 65nm standard cell Verilog technology file: ncverilog testbench.v multiplier_netlist.v tcbn65gplus.v
Provide the output of the NCVerilog test bench dialogue in your report.