

University of Toronto, Faculty of Applied Science and Engineering  
Department of Electrical and Computer Engineering

## **ECE 1387 - CAD for Digital Circuit Synthesis and Layout**

### **Exercise #1 - Simulated Annealing-based Placement and Timing-Driven Negotiated Congestion Routing**

February 2009

J. Anderson

**Assignment Date:** February 6, 2009  
**Due Date:** February 13, 2009 (before lecture begins)  
**Late Penalty:** **-1 mark per day late, with total marks available = 10**

The purpose of this exercise is two-fold: 1) to use an automatic placement tool based on the Simulated Annealing optimization strategy, and to gain some familiarity with the properties of that strategy, in particular, the cooling schedule and cost functions; and, 2) to experiment with the negotiated congestion routing algorithm described in class.

The placement and routing (P&R) tool you will use in this exercise is called “VPR”. It is a tool created here at UofT by Vaughn Betz as part of his Ph.D research in the late 90s. It has been enhanced extensively since then by Professor Rose’s graduate students. As well as being a P&R tool, VPR is a framework for FPGA CAD and architectural research, used by researchers all around the world. VPR takes a text description of the target FPGA architecture as input. The description specifies, among other things, the make-up of the FPGA logic blocks, the routing segment lengths, the switch block style, and the routing delays. By changing the architecture description file, one can evaluate the speed and area-efficiency of different FPGAs.

#### **Access to Software**

The VPR code can be found on the course web page. Un-tar the code, enter the VPR\_HET directory, and type make to compile the code. (I verified that it compiles on EECG Solaris [seth.eecg] and Linux [platinum.eecg]).

#### **Instruction Manual for VPR**

The VPR instruction manual is provided on the course web page. You need only to consult **Section 5** of the manual, which has the parameter information you’ll need. The complete manual describes other aspects of VPR and T-VPACK (a tool that packs LUTs into clusters).

#### **Netlist File and Architecture File to Use**

Use the test circuit files (apex4.net, ex5p.net) and architecture file (k4-n10.xml) provided on the course web page. You may find reading the architecture file interesting, as it gives a description of an FPGA architecture.

#### **Exercise A - Placement and Simulated Annealing**

The VPR program allows the user to set various Simulated Annealing parameters: the starting temperature, the ending temperature, the rate at which the temperature decreases, the number of moves per temperature, and the initial random seed. NOTE: For **all** steps in Exercise A, run VPR with the following parameters: “-place\_only” (prevent routing) and “-place\_algorithm bounding\_box” (minimize bounding box perimeter length).

1. Run the vpr program once, with its default parameters, on the apex4.net, ex5p.net netlists. Plot the score (cost function) versus the temperature. If you turn on the “toggle nets” option, you can see the rat’s

nest of wires become less tangled (use the “-auto 1” option). Indicate on the plots the temperature (roughly) at which placement score “freezes”.

**Note:** to avoid the graphics display, use the “-nodisp” option.

2. Run VPR 5 times for each circuit with different random seeds. Calculate the mean and standard deviation of the resulting final scores. Comment on the sensitivity of the scores to the random seeds; how many runs (with different seeds) are needed so that the mean final score is not changing much? Do both circuits (apex4, ex5p) exhibit the same sensitivity to the random seed?
3. In the SRC directory, in the file `place.c`, at lines 1108-1126, you will see the default annealing schedule used by VPR. Devise a new annealing schedule that produces better scores for both test circuits. Implement your new schedule and recompile VPR. With your new annealing schedule, repeat Step #2 and run VPR 5 times for each circuit, with different random seeds. Report the mean and deviation of the resulting final scores using your new annealing schedule. Include a snippet of C code in your report showing your new annealing schedule. Speculate on why your new schedule yields improved results.

**Note:** In the code, the meaning of the variables is: `success_rat` is the acceptance ratio; `t` is the temperature; `rlim` is range of permitted move distance.

## Exercise B - Timing-Driven Routing

**Note:** For this part of the Exercise, please use the original annealing schedule in VPR and not the new one you devised in Part A.

1. VPR also implements timing-driven routing using an improved version of the PathFinder approach described in class. Run the placement tool with the default options *and* the timing-driven router. Report the post-routing critical path delay. **NOTE:** Do not set “-place\_only” or “-place\_algorithm” for Exercise B. Set the # tracks / channel (W) to 60 using “-route\_chan\_width 60” for apex4.net and set W to 46 for ex5p.net.
2. Using the same parameters as in the previous step, explore the different parameters of the router and try to improve the critical path delay for the two circuits. Comment on the parameters you experimented with and their effect on the critical path delay. Be sure to vary the “present congestion” (“-pres\_fac\_mult”) and “historical congestion” (“-acc\_fac”) parameters talked about in class.
3. For this last question, do not set use the “-route\_chan\_width” parameter. VPR will then automatically find the minimum W needed to route a circuit. Run the placement tool and the timing-driven router 5 times, with different random seeds. Report the average and deviation (across the 5 runs) for two parameters: minimum W and the post-routing critical path delay. VPR’s placer implements a correction factor in its bounding box computation in the `get_net_cost(...)` function in the `place.c` file. The idea is to correct for bounding box underestimates of wirelength for high-fanout nets<sup>1</sup>. Turn OFF the correction faction by setting the variable `crossing` to 1.0 just prior to where `ncost` is assigned. Now, repeat the process of running VPR 5 times, gathering stats on minimum W and critical path delay. Comment on the effect (if any) of the correction factor on W and critical path delay.

---

<sup>1</sup> See reference: Chih-liang Eric Cheng, "Risa: Accurate And Efficient Placement Routability Modeling," *Computer-Aided Design, 1994., IEEE/ACM International Conference on*, pp.690-695, 1994