California Institute of Technology
Department of Computer Science
Computer Architecture

CS184a, Winter 2005

Assignment 6: Interconnect Design

Monday, February 6

Due: Friday, February 18, 9:00AM

For this assignment, you will explore design parameters for mesh networks using Toronto's placement and routing tool `vpr`.

For the two designs given in `/cs/courses/cs184/winter2005/interconnect/nets/` (apex4, tseng): and associated placements in `/cs/courses/cs184/winter2005/interconnect/place/`:

1. Use `vpr` in route only mode to determine how the channel width varies when the segment length is increased from 1 to the row width (36 for apex, 33 for tseng). Make a table and a plot of the results (channel width vs. segment length). You will have a column and curve for each design.

   You don't have to run all 36 points—12 will probably do. Please zero in and provide per track increments around the actual minimum (composite results below), but further from the minimum, you may space samples further apart.

   Keep full population of connection boxes and switch boxes (`Frac_cb` and `Frac_sb` will be 1).

2. Calculate the number of switches per Logic Block for each of the cases. Make a table and a plot of the results (switches per track vs. segment length).

   Pay attention to the Logic Block to channel IO population as described in the architecture file. The Mesh-of-Trees paper shows a picture of this connection (Figure 10).

   I believe `vpr` assumes there is a single switch which allows you to make a corner turn between horizontal and vertical segments which do not end in a particular switchbox. Consequently, each track which does not end at a switchbox contributes 1 switch, while a track that does end contributes 6. (We did not count this extra switch in the equations in the Mesh-of-Trees paper.)

3. Use the previous two results to compute switches per logic block as a function of segment length for each of the designs. Make a table and a plot of the results.

4. Identify the segment lengths which minimize switch requirements for each design. Identify the set of parameters which are within 10% of the minimum. Is there a set of parameters which is in both design's 10% bound?

5. **Bonus/additional thought:** (you probably don't want to do all of these things...but time permitting you might want to think about a few of them.)

    - Can you find a mixed segment length scheme which does better than the homogeneous schemes above? How much?
    - What are the effects of population? (*i.e.* `Frac_cb` and `Frac_sb`) Can you find a substantially lower minimum with partial switchbox/connection box population?
    - Do your results change if you use the fast route option? (*i.e.* if you run the same experiment but using the fast router instead of the quality router, would you have come to the same conclusion about the optimal segment length(s)?)
    - Which segment lengths provide the minimal delay?
    - For your minimum switch count segment length, does switch area or wiring dominate? Does this depend on the number of metal layers? If switches do no dominate, is there a different segment length that actually minimizes area?

vpr and qsub notes:

- You can find a copy of the `vpr` executable on the CS computing systems:
  `/cs/research/ic/develop/vpr/vpr`

- You can find the manual for vpr:
  `/cs/research/ic/develop/vpr_tvpack_manual430.pdf`

- To perform this experiment, you will need to create a separate architecture file for each segment length. The architecture file is described in Section 6.2.3 of the vpr manual (around pp. 17–20). For this problem you will especially be interested in Figure 7 and the example on the top of that page that goes with it.

- A base, segment length 1, architecture file is provided in:
  `/cs/courses/cs184/winter2005/interconnect/arch/seg1.txt`

- A typical command for invoking vpr will look like:

  ```
  vpr apex4.net seg1.txt apex4.placed apex4.seg1.route -nodisp -route_only
  -router_algorithm breadth_first
  ```

  Where:

  - `apex4.net` is the design (the netlist)
  - `seg1.txt` is the architecture file
  - `apex4.placed` is a placement for the design
  - `apex4.seg1.route` is the file in which to store the route for the design
  - `-nodisp` tells it not to bring up the interactive X display
  - `-route_only` tells it not to perform placement (use the given placement)
  - `-route_algorithm breadth_first` tells it to use a channel minimizing router

- If you drop the `-nodisp` option, you can interact with vpr and it will show you pictures of the design. See the vpr manual for further detail on this and other vpr options.

- You will need to make many vpr runs for this assignment. You may want to use the CS Grid Engine to farm your work. See:
  `http://sysadmin.cs.caltech.edu/news/docs/help/software/gridengine_help`
  (It may be necessary to submit jobs from `cs.caltech.edu` rather than from 154 lab machines.)

- A sample script for submitting a single task is given:
  `/cs/courses/cs184/winter2005/interconnect/scripts/vpr.apex4.seg1.example.qsub`. You will need to modify this to point to your own directory (2 places). You will need a different script for each design/architecture combination. One option I've found effective is to create a `Make` file and use it to generate the `qsub` scripts (and perhaps even to submit them). You make also want to use a `Makefile` to generate the various architecture files.

- The standard output from running `vpr` will end up in `cs184-vpr-route.o`NNN, where NNN is the job-id assigned to your job. Note that we specified the file prefix in the sample `qsub` file above; you can change the name there. If you look through this, it will give you statistics on the routing. You will see `vpr` doing a binary search to find the minimum channel width which will support a route. Eventually, it will print a line, like:
  `Best routing used a channel width factor of 12.`
  This is the routed channel width you need. I suggest you look through the full output a few times. However, eventually, you may just want to run:
  `grep Best cs184-vpr-route.o`NNN
  to extract the channel width.