University of Pennsylvania
Department of Electrical and Systems Engineering
Electronic Design Automation

ESE535, Spring 2013 — Assignments #6 and 7 — Monday, March 25

Due: Assign 6: Monday, April 1, beginning of class.
Due: Assign 7 milestone 1: Monday, April 8, beginning of class.
Due: Assign 7 milestone 2: Monday, April 15, beginning of class.
Due: Assign 7 final: Monday, April 22, beginning of class.

Resources You are free to use any books, articles, notes, or papers as references. Provide citations in your writeup as appropriate.

Collaboration You may give tutorial assistance on using OS, compiler, and debugging tools. All code development should be done independently. You may **not** share code or show each other code solutions. All writeups must be the work of the individual.

We will consider coupled projects. However, there should be clear pieces of the project that are the work of each individual. It should be possible to evaluate each piece independently in addition to evaluating the composite effect of the pieces. For example, if you treated assignment 5b as placement on indivisible clusters, then assignment 4 (packing) and assignment 5b (placement) could be considered coupled but independently evaluatable. You could evaluate placement with dumb packings (like sequential place), and packing with some other placement (as we did in assignment 4 or using an external tool like VPR). Then, you could evaluate the composite effect of using good packing and good placement together.

Writeup Turn-in assignments on blackboard. See details on course web page. No handwriting or hand-drawn figures. See details below on what you need to turn in and the format.

Nature of Project Assignment 7 is a more open-ended optimization of your choosing. The deliverable for Assignment 6 is the proposal (essentially an assignment statement) for your Assignment 7. Including the proposal development, you have a total of 4 weeks for this assignment. As a discipline for you and an opportunity for feedback for the instructor, there will be weekly milestones starting with the proposal. You will select the nature of the other two milestones.

The idea here is to take the partially-defective LUT Mapping optimizations further in some way and to some depth. We're leaving it up to you to select how. Some examples:

- Make it more real in some ways – we've deliberately made simplifying assumptions to keep the scope of the assignments down. There are several direction that you would want to take this to better match a realistic component or to explore details of the architecture. Examples include:
  - consider coupled constraints between LUTs when assigning phases (e.g. we previously assumed you could independently select the phase of each input on each LUT, but, without changing the architecture, it might be necessary to select a single phase for each net; that would force all consuming LUTs to use the same phase for a particular net).
  - deal with depopulated and potentially faulty cluster crossbars. We assumed inputs could be permuted arbitrarily. This corresponds to a fully-populated input crossbar, which is not typical of today's, efficient Island-style FPGA clusters [3, 2].
  - model and optimize for hardwired logic connections, such as carry cascades [1] (`http://www.seas.upenn.edu/~ese534/lectures/Day13_6up.pdf`).

  Examples above are intended to be concrete and illustrative. You are welcome to identify other issues that you believe are inadequately addressed by the assignment 2–5 flow for which you could develop suitable models and optimizations.
- Address some piece of the flow we have not tackled in class – we have focused mostly on physical optimizations (clustering, packing placement); you may want to explore some other piece that we've covered in class but not on assignments. Examples include:
  - routing (*e.g.* also perform routing; deal with failures in interconnect as well as logic)
  - logic optimization (*e.g.* avoid generating LUT configurations that have low defect tolerance)
- Develop a better solution to some piece of the flow or a composite optimization that simultaneously attacks multiple parts of the flow – since we only had one or two weeks for each assignment, it was not possible to explore all options and was likely not possible to explore the best approaches for each problem. Furthermore, we have covered more techniques since some assignments were given. You could use this assignment as an opportunity to explore a more aggressive optimization for one of the tasks previously addressed. Examples include:
  - Use SAT, ILP, or pruning search for an earlier task.
  - Approach packing or placement differently.

- Target some utility criteria or cost function that we have not addressed in the course – so far we have focused only on area and wirelength in specific ways for specific assignments. Are there different optimization criteria we might want to optimize? Examples:
  - Should we treat energy more directly?
  - How would we deal with delay directly?
  - How would the mapping change if we were concerned about enhancing robustness to aging? or fast repair in the presence of aging?
  - How fast could you make the per-part mapping (local swapping, placement repair)? Could it be under 10 milliseconds for something as large as `des-em4` or `frisc-em4`?

You are free to consider techniques not introduced in the course, including starting with algorithms from the literature that we did not read for the course. One goal of the course is to enable you to read the literature to follow new ideas as they are published.

Project Formulation Project formulation must include:

- Defining the (potentially revised) architecture model
- Defining the final evaluation cost function
- Defining the optimization task and goal
- Defining any new solution legality checking code for your revised architecture model
- Defining the schedule and milestones
- Defining the evaluation experiments (including benchmark set and targets or parameters to the optimization problem)

Your Assignment 6 project formulation will be much like the assignments we have been providing you for assignments 2–5. You are, in essence, creating an assignment for yourself for Assignment 7. Reviewing assignments 2–5, you will see that all of the above items are defined in them (with assignments 2a and 5a being simple milestones). Being able to formulate problems is an important skill to develop and one of the goals of this course.

Milestones Since Assignment 7 is a three week project after you turn in Assignment 6, you should identify what you should target completing at each of the intermediate two weeks. What exactly the milestones are will be project dependent, but they should be related to development of the code and evaluation of solutions. Some things you might think about when selecting milestones:

- is there support code (e.g. cost function calculations, legality checks, modifications to key data structures) that should be done early? (during the first week)?
- are there examples you should work by hand first? (like the warmup exercise you did for earlier assignments)
- is there a very simple version you can get running early as a baseline or starting point? (particularly if you are attacking a piece of the flow we did not target in class – this might be similar to doing the legal placement for assignment 5a before doing the wirelength-minimizing placement of assignment 5b).

- are there experiments you need to run early to help decide how to tune your algorithm?
- perhaps you could target a running implementation by the second week, giving you time to tune and improve it during week 3?

Feedback We will make an effort to get you rapid feedback on Assignment 6.

Algorithm Selection and Tuning As a longer, more open-ended project, you should be exploring alternative algorithms/approaches and tuning the parameters (*e.g.* cooling scheduling in simulated annealing, weights in optimization cost functions). Your final writeup should describe what you explored and what you learned. This may include additional result graphs comparing algorithms and parameters. A good example of an article showing the exploration of tuning parameters is [4]; notably Tables 1.1–1.5 and Figure 4 show the impact of various parameters in their algorithm.

Turnin

Assignment	Points	Pieces	Description
6	5	PDF only	As identified above under **Project Formulation**, like an assignment statement
7 milestone 1	3	PDF and code	As you define in Assignment 6
7 milestone 2	3	PDF and code	As you define in Assignment 6
7 final	14	PDF and code	Writeup will likely end up looking similar to your answers to assignments 2b, 3, 4, 5b with an additional section on **Algorithm Selection and Tuning** as described above.

References

[1] Kevin Chung and Jonathan Rose. Tempt: Technology mapping for exploration of fpga architectures with hard-wired connections. In *Proceedings of the 29th ACM/IEEE Design Automation Conference*, pages 361–367. IEEE, June 1992.

[2] Kunihiro Fujiyoshi, Yoji Kajitani, and Hiroshi Niitsu. Design of minimum and uniform bipartites for optimum connection blocks of fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 16(11):1377–1383, November 1997.

[3] Guy Lemieux and David Lewis. Using sparse crossbars within LUT clusters. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 59–68, 2001.

[4] Alexander Marquardt, Vaughn Betz, and Jonathan Rose. Timing-driven placement for FPGAs. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 203–213, 2000.