UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
CPEN 513: CAD Algorithms for Integrated Circuits
2020/2021 Term 2

Final Project

Final project due April 14, 2021, 11:59pm

The project for this course is very open-ended. You can do anything you want, as long as it has something to do with the course material. This might include:

- Implementing and evaluating a known algorithm
- Comparing two or more existing algorithms
- Improving an existing algorithm
- Investigating the effects of various parameters on an algorithm's performance
- Coming up with an entirely new algorithm to solve some problem
- Applying an existing algorithm to a different problem
- Investigating interactions between the target technology and a CAD tool
- A theoretical analysis of some algorithm

You can work in groups of anywhere from 1 to 3 people. As a rule-of-thumb, the level of difficulty expected is:

Number of people * level-of-difficulty-for-each-assignment * 2

If you work in a group, be sure to carefully partition the work so that the contribution of each student is clear.

The deliverable for this part of the project will be a report outlining what you have done, the results, as well as the code you have written. The nature of this report will depend very much on what your project is. It may describe a reimplementation of an existing algorithm, or it may describe a new algorithm that you develop. You can present your results any way you like. If you are implementing an algorithm, some graphics would be nice. If you are comparing algorithms, some tables showing comparative results would be good. It is entirely up to you.

Note: it is ***not*** expected that you compare the results of your algorithm to those in previous work, however, a very strong project likely would contain at least some comparison to the literature.

In order to ensure that everyone has a suitable project, I am asking you to submit a topic by March 11th by email to stevew@ece.ubc.ca. You should indicate the project title and the members of your group (if you are not working on your own). You should also write enough to convince me that the project is interesting, and is of a suitable difficulty. If multiple groups/people are working on the same project topic, that is ok.

If you are working on a project that you think might result in an interesting research paper, please let me know as soon as possible. For projects related to FPGAs, consider the International Conference on Field-Programmable Technology (deadline is June) or the International Conference on Field-Programmable Logic and Applications (deadline is mid-March). There is also the International Conference on Computer Design (deadline also in March) and the International Conference on Computer-Aided Design (deadline in April; this conference is very competitive). Note that you don't need to do a project worthy of publication for this course; this is only for the keeners. But it is very possible; two years ago, a student published his project in a conference in Dublin, and then presented a demonstration of the work at a conference in Japan. In 2011, a student published a paper related to his project in the International Conference on Reconfigurable Computing and went to Cancun to present it. In 2009, a student published his final project as a research paper at the International Conference on Field-Programmable Technology in Sydney, Australia, and in 2006, a student published a paper based on his course work and went to a conference in Bangkok to present it. Unfortunately, this year, international travel may be out, but most conferences are still being held virtually.

Here are some ideas, *but they are only suggestions*. I would prefer that you come up with an idea on your own. Other ideas will be given thoughout the course. Please feel free to come up with your own ideas as well. **If you want to do an idea related to your thesis research**, it is ok, as long as there is a clear

distinction between what counts towards your CPEN 513 project and what counts towards your thesis. If you are going this route, talk to me (and your thesis supervisor) early.

1. Machine Learning is receiving much attention. The community is still trying to understand how machine learning (deep learning, reinforcement learning, etc) can be used in CAD. Choose an algorithm and investigate whether machine learning is a viable candidate. I would suggest building on top a package like Tensorflow or similar so you don't have to implement the low-level details. Be careful, because this project could expand to fill too much time.
2. Create a genetic algorithm-based placer, router, partitioner, or packer.\
3. In class, we will talk about an embedding algorithm for a Quantum Annealer. The algorithm we will talk about has been evaluated on a previous-generation Quantum Computer from DWave. As we will talk about in a video, DWave has a new architecture called Pegasus. Re-implement the embedding algorithm for Pegasus. If you like, you can come up with your own embedding algorithm.
4. Current CAD tools require a large amount of run-time memory. Consider implementing one of the algorithms we talked about, but optimize the run-time memory usage. Here, I am looking for more than just efficient data structures; perhaps that is something you can do in the algorithm which minimizes the amount of memory required.
5. ISPD 2021 has a contest related to Wafer Scale Modeling. A description of the contest is at https://www.cerebras.net/ispd-2021-contest/ If you would actually like to enter the contest, you will need to have the project done by the contest deadline of March 10. If you are doing this, please contact me early to see if we can find a re-arrangement of the course deliverables to allow you to work on your project first. You can still work on the project after March 10th, but you won't be able to enter the contest. Note that this will be a fair bit of work, and you may want to work with a team.
6. Something for those who want more "hands-on" experience with CAD Tools: Modern CAD tools typically "mangle" net or module names, often to avoid special characters (eg. i[0] may become i_0_ internally). Mapping back to the original names may be easy for a human, but sometimes more difficult to do in an automated way. The project is to build a "reverse mapper" as described http://iccad-contest.org/2018/Problem_A/2018ICCADContest_ProblemA.pdf possibly using machine learning techniques. Test cases can be found here http://iccad-contest.org/2018/problems.html (Problem A).
7. In class, we will talk about a method of reducing the memory footprint of the Routing Resource Graph inside an FPGA. Another interesting possibility is to minimize the size of the structure used to store the circuit representation. It would be more than just efficient coding; if you can find a fundamental change in the way circuits are represented, it would likely be a publishable paper.
8. Invent a new packing algorithm (packing four-input lookup tables into clusters in an FPGA). Compare your results to those obtained using existing tools.
9. Hardware acceleration can be used to speed up a CAD tool. Consider building an accelerator for a maze router.
10. In Lecture 1, we talked about Bus-oriented routing. Implement the algorithm we saw in Paper Review 1, or implement an algorithm of your own choosing to solve a similar problem. This project was motivated by http://iccad-contest.org/2018/Problem_B/2018ICCADContest_ProblemB.pdf (test cases at http://iccad-contest.org/2018/problems.html)
11. A more theoretical project: when compiling a circuit multiple times, it is often desirable to use previous compilation results to guide the new compilation. A key requirement when doing this is figuring out how to determine the "difference" between the current circuit and the circuit compiled previously. Define a measure of "difference" between two circuits, and write a CAD tool to compute the difference between these two circuits.
12. There are several analytical models that describe the expected result of a CAD tool. Pick one of these models and perform experiments to see how well the model describes reality. If you have time, see if you can improve the model.
13. Create a fast partitioner. Very fast. Faster than F&M. Quantify how much worse the results are and how much faster than F&M it goes. Have some sort of parameter, so the user can trade-off quality for speed.
14. Create a program to calculate the (partitioning) Rent parameter of a circuit (probably using recursive bi-partitioning). Determine how consistent the Rent parameter is as a function of how you partition your circuit. Alternatively, use a placement tool and calculate the placement Rent parameter. Perhaps see if the Rent parameter changes for various types or sizes of circuits.

15. Investigate how an FPGA architecture affects the quality of solutions produced by place and route tools (say, VPR). Consider the connection block flexibility and the pattern of the switches in the connection block.
16. Implement a fast-placement algorithm. Maybe you want to cluster and then place.
17. Add a really intelligent rip-up and re-route scheme to a maze router. Try it on larger circuits and see how it works compared to an intelligent net re-ordering scheme.
18. Use either a commercial HLS tool (eg. Vivado HLS) or an academic HLS tool (eg. LegUp) to implement a complex application, and optimize it for an appropriate metric. One suggestion would be a multi-layer neural network (inference). If there is a group, it would be interesting to also implement on-chip training.