CSC2547 Project Proposal

Guanglei(Gabriel) Zhou, Yufei Kang, Zhaoyang (Tommy) Zhu
University of Toronto

March 13, 2021

Abstract

The emphasis of our project is in the application area, where we will investigate the phase-ordering problem in logic synthesis and utilize Reinforcement learning techniques to optimize its performance. In the following sections, we will describe the background and prior work, define the problem, and state our tentative goal for this project.

1 Background

In computer engineering, Logic synthesis is the process that transforms the two of these abstraction levels named register-transfer level (RTL) and transistor gate level. It is a highly automated process that takes the digital design in RTL and transforms it into an optimized transistor-gate representation. In the process of logic synthesis,there exists a set of optimization algorithms such as balancing,rewrite, refactoring, etc.. The permutation of such commands can lead to different result. So how to decide the right sequence of such optimizations commands for a input circuit has become the bottleneck for the logic synthesis optimization.

2 Prior Work

In recent years, some researchers have begun investigating the use of RL to determine good orders for applying optimization passes in high-level synthesis [1] and deciding chip placement[2]. These works demonstrate the promising benefit of using reinforcement learning techniques in the CAD algorithms. For logic synthesis, DRiLLs [3] framework is the first work that combines the CAD in Logic synthesis with RL agent. It uses the RL agent to decide the best order of applying the optimization commands.However, it is only implemented the A2C algorithm and by reproducing their result, no clear sign of learning is observed for the RL agent. Thus, we will use this work as a basis, and improve it using what we have learned in class.

3 Problem Definition

Use RL agent to decide the best sequence of optimization commands in the logic synthesis CAD so that an adaptive optimization for different input commands can be achieved. The actions and states will be inherited from DRiLLs [3] framework set up and they are both discrete. The reward will be coming from the reduction of area / latency cost on hardware, and the environment will be the logic synthesis CAD tool. Also, since getting a sample is rather expensive in the logic synthesis CAD, so the maximum available samples for any input circuits will be limited to around 2500 sample points. Sample efficiency will be a great burden in this application area and should be aware of in the lateral part of this project. The work flow of this project includes,

1. Improve previous DRiLLs framework to GYM-Style Environment

2. Explore different RL algorithms and different hyper parameters to explore the performance gain

3. Try out Linear Function Approximation like tile encoding to help with the lack of samples issue. (Optional, if time allows)

The first step is essential since after packing the logic synthesis cad tool in the GYM-style environment, it gives us access to RL libraries so that we can save some time writing the RL code all by ourselves. For the second step, we plan to have each group mate responsible for one of the RL algorithms and observe its performance. There will be two targets for this project. Firstly, achieve a similar and even better performance with the prior work using A2C. Secondly, found a clear learning sign for the RL agent under the constrain that only thousands of samples are given.

References

[1] Ameer Haj Ali, Qijing Huang, William Moses, John Xiang, Ion Stoica, Krste Asanovic, and John Wawrzynek. Autophase: Compiler phase-ordering for high level synthesis with deep reinforcement learning. *CoRR*, abs/1901.04615, 2019.

[2] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Sungmin Bae, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Anand Babu, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter, and Jeff Dean. Chip placement with deep reinforcement learning, 2020.

[3] A. Hosny, S. Hashemi, M. Shalan, and S. Reda. Drills: Deep reinforcement learning for logic synthesis, 2020.