CS516000 FPGA Architecture & CAD

Final Project Report

111062684　林鼎勳

I. Introduction

In this project, we are asked to partition the nodes of a given circuit into several FPGAs to minimize the sum of external degrees of all FPGAs subject to the following three constraints: 1) Fixed node constraint, 2) Partition size constraint, and 3) Topology constraint.

II. Algorithm

The overall concepts come from [1]. But there are some differences between the paper and this project, I modify some parts of it.

My algorithm can be divided into three stages: propagation, partition, and violation. The remaining part will describe them.

A. Propagation:

In this part, I will choose all the fixed node-FPGA pairs and put them in their desired FPGA. At the same time, I will put the adjacent nodes into the waiting queue and figure out their candidate FPGAs.

B. Partition:

Here, I have a waiting queue from the previous stage. Then I will choose the node with the least number of candidate FPGAs first and calculate the increase of the external degree of different candidate FPGAs. After calculating, I will choose the one with the least increase of external degree and put it to the desired FPGAs. Later, the adjacent nodes of this determined node will be put into the waiting queue. The process will go on and on until the waiting queue is empty.

During this process, there might exist a situation in which the candidate FPGAs of a node is empty, so the node has no FPGA to go. If this happens, I will put it into a list and skip it. I will handle these undecided nodes in the next stage.

[1] D. Zheng, X. Zang and M. D. F. Wong, "TopoPart: a Multi-level Topology-Driven Partitioning Framework for Multi-FPGA Systems," 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2021, pp. 1-8, doi: 10.1109/ICCAD51958.2021.9643481.

C. Violation

There might be some situations that the program cannot put all the nodes into a place without any violation. So, this part will handle the node remains from the previous part.

After collecting all the nodes, I will make a list of all non-full FPGAs. Then I calculate all combinations to find a minimum cost combination.

III. Results

These are my results running on the Matlab server:

Case	Total Runtime	Topology Violation	Total External Degree
B1	0.0008 sec.	0	36
B2	0.0020 sec.	0	160
B3	0.0122 sec.	0	430
B4	0.0209 sec.	9	1135
B5	0.2434 sec.	33	11295
B6	7.12 sec.	130	61575
B7	30.75 sec.	369	121329
B8	119.10 sec.	635	245740

The results show that in smaller node sizes, my program can produce a zero-violation result. However, with the increasing number of nodes, it comes out some nodes cannot be put into a non-violation FPGA. As for the timing part, all cases are under 10 minutes constraint. The longest time I need to spend on the largest case has cost only 2 minutes.

IV. Conclusion

In this project, I learned the concept of topology-driven partition. Under this partitioning strategy, we can reduce the length of the net because all adjacent nodes will try to put together or put in a neighborhood place. Besides, I also met lots of difficulties in this project, like how to handle a node if it doesn't have a non-violation place or how to decide the priority of putting the node into FPGAs.