CS 5160-00 FPGA Architecture & CAD Fall 2018

Programming Assignment
Due: Nov 10, 2018

K-LUT Technology Mapping

1 Background

The combinational part of a circuit can be represented by a directed acyclic graph (DAG). Each node in the graph corresponds to a logic gate, and each edge represents a signal wire in the logic circuit. A primary input node has no incoming edge, and a primary output node has no fanout edge. For a sequential circuit, we can break it into several DAGs while considering the inputs and outputs of registers as additional POs and PIs.

2 Problem

In this assignment, you have to **write a C/C++ program for K-LUT technology mapping that optimizes the depth and the area**. The depth is computed by the maximum number of levels of K-LUTs from a PI to a PO and area is given by the number of LUTs used. You are free to adapt any algorithm introduced in the class or reported in the literature or design your own algorithm.

3 Evaluation

1. For each case, if the mapping solution is not valid, you will get **0** score on that test case, otherwise it is determined by the sum of the area score and the depth score.
2. The area (depth) score is based on the area (depth) of your mapping result compared to other students when your solution is valid. Smaller area (depth) gets higher area (depth) score. Here is the equation for area score calculation:

$$50\% - 10 \times \left(\frac{\text{Your area}}{\text{The smallest area among all teams}} - 1 \right)$$

3. Additionally, if your program takes more than **2** minutes to generate a mapping result, it fails on that test case002E

4 Input format

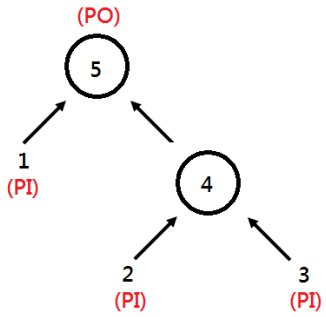
Your program should be invoked using a command in the format below where K is the LUT size:
./program_name input_file_name K
e.g. ./map alu4.aag 4

The input file describes a directed acyclic graph (DAG), and all the nodes in the DAG have exactly two inputs except the PI nodes. The first line contains the string “agg” followed by three integers, *m,*

i, and *o*. *m* denotes the total number of nodes in the DAG. *i* denotes the number of primary input nodes (PIs). *o* denotes the number of primary output nodes (POs). The following *i* lines list the node ID of each PI, and the next *o* lines list the node ID of each PO. The remaining lines provide the fanin information of each non-PI node where each line contains three integers. The first integer is the ID of a non-PI node, the second and third integers are the IDs of the two fanins of this node.

Here we show the content of a sample input file "simple.aag":

```
agg 5 3 1 //m = 5, i = 3, o = 1
1          //PI
2          //PI
3          //PI
5          //PO
4 2 3      //node 4 with two fanins 2 and 3
5 1 4      //node 5 with two fanins 1 and 4
```

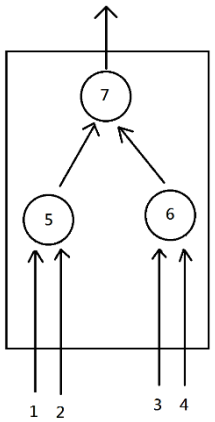


5 Output format

The output file format is described below. Every line in the output file represent a LUT. The first integer is the output node ID of the LUT, and the remaining integers are the IDs of all the fanin nodes of the same LUT.

For example, if the mapping contains a LUT with 4 input nodes as shown in the figure below, then the output line would be like

```
7 1 2 3 4      //first one is the output node
               //and the following are the input nodes
```



(Note that the order of the input nodes is not important, so "7 3 1 2 4", "7 4 3 2 1", etc. are also acceptable.)

6 Project Submission

Source codes should be uploaded to iLMS. Please include a Makefile for compiling your codes and a Readme file for relevant information in using your codes. You also need to submit a report which describes and analyzes the algorithm you used.

7 Environment Issue

The official evaluation platform will be an Ubuntu workstation, gcc and g++ compilers are available. We do not use Windows platform for evaluation, so please include your Makefile.