CS6135 VLSI Physical Design Automation

Homework 2: k-way Min-cut Partitioning

Due: 23:59, October 21, 2025

1. Introduction

Let $\boldsymbol{C}$ be a set of cells and $\boldsymbol{N}$ be a set of nets. Each net connects a subset of cells. The k-way min-cut partitioning problem is to partition the cell set into k disjoint groups $\boldsymbol{A}$, $\boldsymbol{B}$, ⋯. The cost of a k-way partitioning is measured by the cut size, which is the number of nets having cells in multiple groups. In this homework, you are asked to implement and modify Fiduccia-Mattheyses (FM) algorithm to solve the problem of two-way/four-way min-cut partitioning.

2. Problem Description

In this homework, you are asked to partition all cells into two and four groups each of which is put in a different group without exceeding the balance factor. The k-way min-cut partitioning problem is defined as follows:

(1) Input:

- ✓ Cell and its size
- ✓ Netlist

(2) Output:

- ✓ The final cut size and a partitioning result

(3) Objective:

Partition the circuit in multiple groups $\boldsymbol{A}$, $\boldsymbol{B}$, ⋯, such that the cut size is minimized subject to the following condition.

For each group G in two-way min-cut partitioning:

$$0.45 \leq \frac{\textbf{size}(\boldsymbol{G})}{\textbf{size}(\boldsymbol{Total})} \leq 0.55$$

For each group G in four-way min-cut partitioning:

$$0.225 \leq \frac{\textbf{size}(\boldsymbol{G})}{\textbf{size}(\boldsymbol{Total})} \leq 0.275$$

Here $\textbf{size}(\boldsymbol{G})$ is the sum of all cell sizes in group $\boldsymbol{G}$ and $\textbf{size}(\boldsymbol{Total})$ is the total cell size.

3. Input File

(1) The *.txt* file:

The .txt file specifies the input information. Here is an example:

```
NumCells 11
// NumCells number of cells
Cell C1 13
// Cell cell name cell size
      ⋮

NumNets 12
// NumNets number of nets
Net N1 18
// Net net name number of cells on the net
Cell C1
// Cell cell name
      ⋮
```

4. Output File

(1) The *.out* file:

It reports the cells in each group and the cut size. For two-way min-cut partitioning, the group names are "GroupA" and "GroupB"; for four-way min-cut partitioning, the group names are "GroupA", "GroupB", "GroupC", and "GroupD". You can run the "verify" program to check whether your result is legal or not. Here is an example:

```
CutSize 1
// CutSize cut size
GroupA 4
// GroupA number of cells in GroupA
C1
// cell name
      ⋮
GroupB 4
// GroupB number of cells in GroupB
C2
      ⋮
```

5. Language/Platform

(1) Language: C/C++

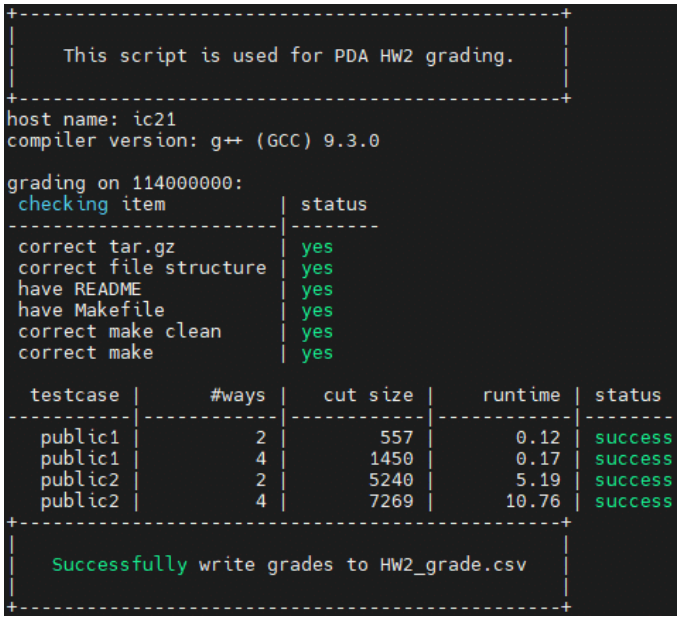
(2) Platform: Unix/Linux

6. Report

Your report should contain the following content, and you can add more as you wish.

(1) Your name and student ID

(2) The final cut size and the runtime of each testcase. Paste the screenshot of the result of running the **HW2_grading.sh** as the picture shown below.



(3) The details of your implementation containing explanations of the following questions.

I. Where is the difference between your algorithm and FM Algorithm described in class?

II. Did you implement the bucket list data structure?

- ✓ If so, is it exactly the same as that described in the slide? How many are they?
- ✓ If not, why? You had a better data structure? Or, is bucket list useless?

III. How did you find the maximum partial sum and restore the result?

IV. How did you modify the FM algorithm to handle four-way min-cut partitioning? You could use flow chart(s) and/or pseudo code to help elaborate your algorithm.

(4) What have you learned from this homework? What problem(s) have you encountered in this homework?

7. Required Items

Please compress `HW2/` (using tar) into one with the name `CS6135_HW2_${StudentID}.tar.gz` before uploading it to eeclass.

(1) `src/` contains all your source code, your `Makefile` and `README`.

- `README` must contain how to compile and execute your program. An example of `README` is like the following figure:

```
--How to Compile
  In "HW2/src/", enter the following command:
  $ make
  An executable file "hw2" will be generated in "HW2/bin/".

  If you want to remove it, please enter the following command:
  $ make clean

--How to Run
  Usage:
  $ ./hw2 <input file> <output file> <number of partitions>

  E.g., in "HW2/bin/", enter the following command:
  $ ./hw2 ../testcase/public1.txt ../output/public1.2way.out 2
```

(2) `output/` contains all your outputs of testcases for the TA to verify.

(3) `bin/` contains your executable file.

(4) `CS6135_HW2_${StudentID}_report.pdf` contains your report.

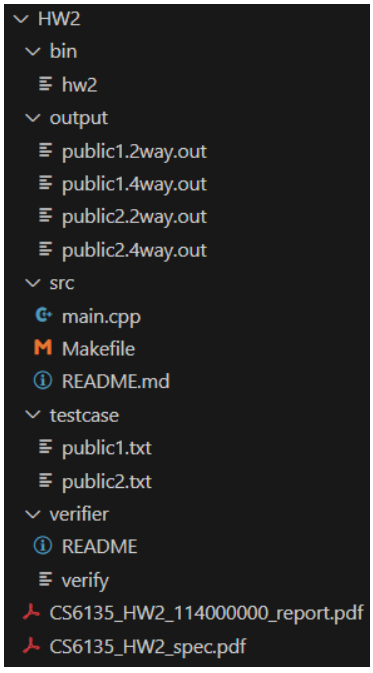
You can use the following command to compress your directory on a workstation:

```
$ tar -zcvf CS6135_HW2_${StudentID}.tar.gz <directory>
```

For example:

```
$ tar -zcvf CS6135_HW2_114000000.tar.gz HW2/
```

The folder structure would be like the following figure:



8. Grading

- ✓ 50%: Solution correctness. For public and hidden testcases, you need to generate a valid result.
- ✓ 30%: Solution quality. The solution quality (final cut size) of each hidden testcase.
- ✓ 20%: The completeness of your report

P.S. Using C++11, C++14, or C++17

The C++11 standard is implemented in GCC 4.8.1 and beyond. If you want to use C++14 or C++17, you need to use GCC 6.1 and beyond.

```
[chlu19@ic51 ~]$ g++ --version
g++ (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

If you want to change the GCC version, you can follow the news of the login information and change your GCC version by yourself.

```
--------------------NTHU CS VLSI/CAD News--------------------
.If you encounter problems with Python3, please use command
    "setenv LD_LIBRARY_PATH "/lib64/":$LD_LIBRARY_PATH".
.For gcc 4.8.5 on centos 5 or gcc 4.9.3 on centos 6, use command
    "source /tools/linux/gnu/setup_toolkit.csh".
.For gcc 9.3.0 on ic21, ic22, ic51, and ic55, use command
    "source /tools/linux/gnu/setup_gcc_9.3.0.csh".
.For loading information, use command "lab_uptime".
.For platform information, use command "lab_plat".
.For apply new account, please fill this sheet:
    https://bit.ly/2FGbnMg
.If you have any problem, please contact us:
    nthucad.cs@gmail.com
.Please read this FAQ.
    http://nthucad.cs.nthu.edu.tw/~webster/CADWorkstationFAQ.html
------------------------------------------------------------
```

In this way, you need to source it every time when you log in on a server. Instead, you can create a shell resource file called `.tcshrc` in the root folder and put the source command in it. Just enter the following command once, re-login, and then it will never bother you anymore.

```
$ echo "source /tools/linux/gnu/setup_gcc_9.3.0.csh" >> ~/.tcshrc
```

P.S. Using Boost C++ Library

The boost C++ library is installed on ic21, ic22, and ic51. If you want to use boost C++ library, you must add the following include path while compiling your source code.

```
-I /usr/local/include/boost/
```

For example:

```
$ g++ -O3 -std=c++11 -I /usr/local/include/boost/ main.cpp -o hw2
```

Notes:

- Make sure the following commands can be executed.
  - Go into directory "`src/`", enter "`make`" to compile your program and generate the executable file, called "`hw2`", which will be in directory "`bin/`".
  - Go into directory "`src/`", enter "`make clean`" to delete your executable file.
- Please use the following command format to run your program.

  ```
  $ ./hw2 <input file> <output file> <number of partitions>
  ```

  E.g.:

  ```
  $ ./hw2 ../testcase/public1.txt ../output/public1.2way.out 2
  ```
- Use arguments to read the file path. Do not write the file path in your code.
- Program must be terminated within 100 seconds for each testcase.
- Please use ic21, ic22, or ic51 to test your program.
- We will test your program by a shell script with GCC 9.3.0 on ic21. Please make sure your program can be executed by **HW2_grading.sh**. If we cannot compile or execute your program by the script, you will get 0 points on your programming score.
- Note that any form of plagiarism is strictly prohibited, including the code found on GitHub and the code from any student who took this course before. If you have any problem, please contact TA.
- Notes on AI Tool Usage
  (1) Do not submit code generated directly by AI or rewritten from public repositories.
  (2) You may use AI for debugging or code improvement, but you must disclose how it was used.
  (3) Clearly state in comments or a separate note which AI tools you used and for what purpose.
  (4) You are responsible for verifying correctness and being able to explain your final code.