Andy, Yu-Guang Chen

EE6094 CAD for VLSI Design
Programming Assignment 3 (Due: 23:59:59, 05/12/2022)

Introduction

In the front-end design stage, designers synthesize the RTL design into the corresponding gate-level netlist and then move on to physical design stage. In the first step of physical design stage, designers need to separate the entire gate-level netlist into several sub-blocks so that each sub-block can be placed based on floorplanning results. This process is called **Partitioning**. The goal of partitioning is to split the gate-level netlist into several sub-blocks such that the number of connections between sub-blocks is minimized.

Background

In integrated circuit design procedure as shown in Figure 1, physical design is a step in the standard design cycle which follows after the circuit design. At this step, circuit representations of the components (devices and interconnects) of the design are converted into geometric representations of shapes which, when manufactured in the corresponding layers of materials, will ensure the required functioning of the components. This geometric representation is called integrated circuit layout. This step is usually split into several sub-steps, which include both design and verification and validation of the layout.

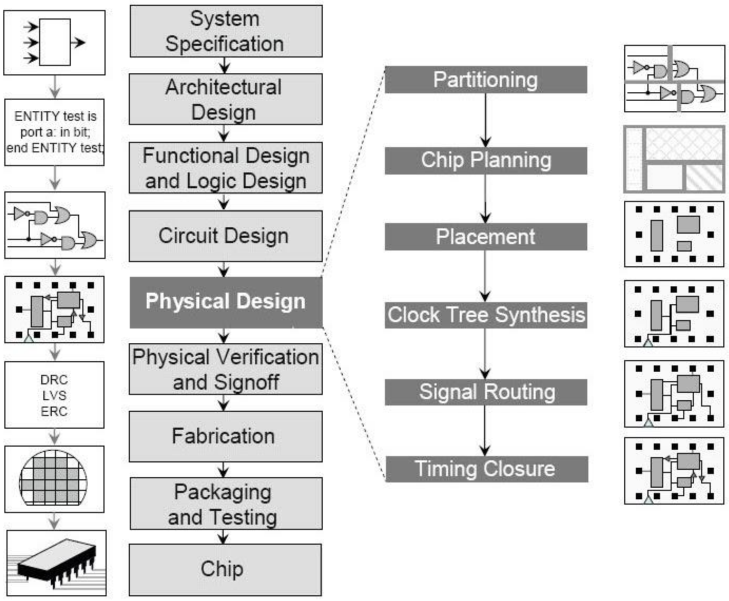


Figure1. VLSI Design Procedure

Circuit partitioning plays an important role in the physical design phase. The objective is to partition the circuit into parts such that the sizes of the components are within prescribed ranges and the complexity of connections between the components is minimized. Therefore, a good circuit partition can make other procedures in physical design easier to generate better results.

Problem Definition

Let $C = \{C_1, C_2\,, \ldots\, C_n\}$ be a set of cells and $N = \{N_1, N_2, \ldots, N_m\}$ be a set of nets. Each net N_i, $i = 1, 2, \ldots, m$, connects a subset of cells. The two-way min-cut partitioning problem is to partition the cell set into two groups *A* and *B*. The size of group *A*, *size*(*A*), is the number of cells in group *A*. *Size*(*B)* is the number of cells in group B. The cost of a two-way partitioning is measured by the cut size, which is the number of nets having cells in both groups. The problem is defined as follows:

Input: A net-list for a circuit.

Output: Partition the circuit to two sub-circuits *A* and *B* so that the cut size of sub-circuits *A* and *B* is minimized.

Constraints:

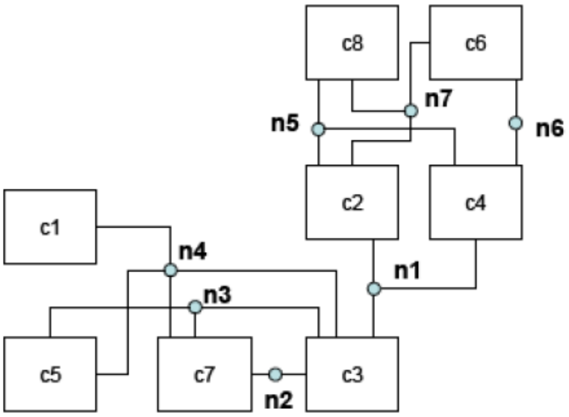
1. $|size(A) - size(B)| < n/5$, where *n* is the number of cells in the circuit.
2. The runtime of your program is limited to at most 1 hour per testcase.

Input file format

Input is a list of nets. Each net statement starts with the keyword “NET” and the name of the net. The cells that are connected by the net are listed between a pair of braces following the net name. The net may be provided in random sequence (ex: n1, n7, n2, n5 …).

Example:

```
NET n1 {c2 c3 c4}
NET n2 {c3 c7}
NET n3 {c3 c5 c7}
NET n4 {c1 c3 c5 c7}
NET n5 {c2 c4 c8}
NET n6 {c4 c6}
NET n7 {c2 c6 c8}
```



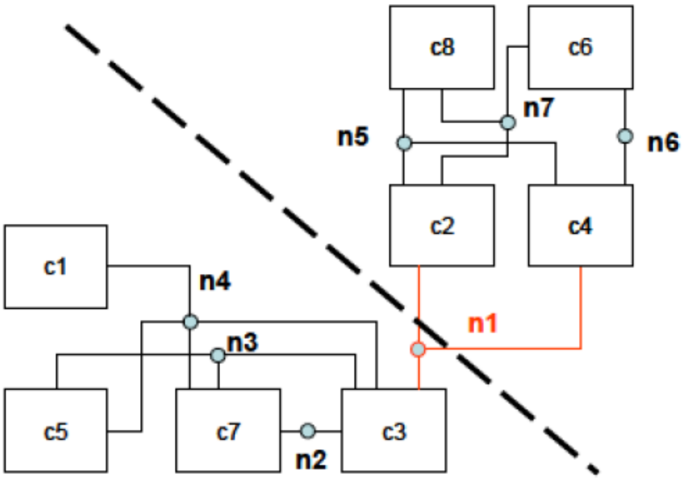
Output file format

Report the cells in each group and the cut-size. **Please follow the output format.**

```
cut_size #
A
cell_ID
…
B
cell_ID
…
…
```

Example:

```
cut_size 1
A
c1
c3
c5
c7
B
c2
c4
c6
c8
```



Algorithm

You are asked to implement **Fiduccia-Mattheyses Algorithm** to solve the problem of two-way min-cut partitioning. You are not allowed to use another algorithm to solve the problem.

Requirement

1. You have to write this program in C or C++. We will verify your program on a workstation (the same one for your PA1 & PA2). No open source codes are allowed to use. (i.e., you MUST implement the tool by yourself).

2. All files should be submitted through ee-class. You have to submit **a source code file** named as *StudID*_PA3.cpp (ex: 9862534_PA3.cpp) and **a report** named *StudID_Name*_PA3_report.pdf (ex: 9862534_陳聿廣_PA3_report.pdf). Note that the only acceptable report file format is .pdf, no .doc/.docx or other files are acceptable. **BE SURE to follow the naming rule mentioned above. Otherwise, your program will be not graded.**

3. The output file of your program should be named as *name***.out** where *name* is the input file name. You also need to write a makefile to compile and run your program. We will verify your program with the makefile your provided. Please see the "Makefile" section for more details.

4. We don't restrict the report format and length. In your report, you have to at least include:
   (1) How to compile and execute your program; (You can use screenshot to explain)
   (2) The completion of the assignment; (If you complete all requirements, just specify all)
   (3) The results and runtimes for all test cases; (You can use a table or a figure to show)
   (4) The hardness of this assignment and how you overcome it;

(5) Any suggestions about this programming assignment?

Makefile

Your makefile should at least contain these 3 commands, which is (1) **make all**, (2) **make run**, and (3) **make clean**. The descriptions of each command is shown below.

(1) **make all**: This command will automatically compile your source codes and generate the corresponding objects and executable file.

(2) **make run**: This command will execute your executable file and run your program.

(3) **make clean**: This command will automatically remove all the objects and executable file generated by **make all**.

Grading

The grading is as follows:

(1) Correctness of your solutions: 30%
(2) Quality of your solutions: 20%
(3) Readability of your code: 10%
(4) The report: 10%
(5) Demo session: 30%

Please submit your assignment on time. Otherwise, the penalty rule will apply:

- Within 24hrs delay: 20% off
- Within 48hrs delay: 40% off
- More than 48hrs: 0 point

Contact

For all questions about PA3, please send E-mail to TA 黃柏燁 (alec2515@gmail.com)

Reference

[1] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions," 19th Design Automation Conference, 1982, pp. 175-181, doi: 10.1109/DAC.1982.1585498.