ENEE640 Final Project–Thermal-aware Floorplanning

Due **Wednesday 05/18/2016**

1 Objective

The placement of architectural blocks, or a particular floorplan selected for a given chip may affect the maximum temperature of the chip considerably. Higher temperature may signicantly affects chip's reliability, increases leakage power and makes cooling solutions more expensive. As a result, thermal-aware floorplanning is urgently needed. In this exam, you will write a thermal-aware floorplanning program. The main goals of this project are:

- To get familiar with the thermal models used in the literature. Most notably, understand the block model used in [1].
- To write a thermal-aware floorplanning program. That is, your program can generate a floorplan of a given circuit that results in a low maximum chip temperature.
- To explore the trade-off between the maximum chip temperature and chip area.
- To write a report based on your experience

2 Problem Statement

Floorplanning is the first stage of the VLSI layout. It is the problem of placing a given set of functional blocks in the plane to minimize an objective function (e.g. maximum chip temperature in this exam) under some constraints. In this exam, we consider all functional blocks to be **hard rectangle blocks**, meaning that they are rectangular with a fixed width and height. They can not be rotated either. You will be given the dimension (width and height) of each functional block, together with its power profile. You are asked to write a floorplanning program that can generate a floorplan of the given circuit that results in a low maximum chip temperature. **You can use whatever programming language you are comfortable with.** The formal description of the problem is given below.

Let $\mathcal{B}$ be a set of n functional blocks: $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$. Let w_i and h_i be the width and height of block b_i respectively. Let A_{sum} denote the sum of area of these blocks: $A_{sum} = \sum_{i=1}^{i=n} w_i \times h_i$. A floorplan of $\mathcal{B}$ is characterized by a set of coordinates $\mathcal{F} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where (x_i, y_i) is the coordinate of the bottom-left corner of block b_i. Note that since each block's width and height cannot be changed, nor can its orientation be changed, $\mathcal{F}$ is sufficient to determine the floorplan of $\mathcal{B}$. We further define the chip area, denoted by A_{chip}, as the area of the smallest rectangle that encloses all blocks in $\mathcal{B}$.

Let p_i denote the power of block b_i, assuming that its power consumption keeps unchanged. Let $\mathcal{P} = (p_1, p_2, \ldots, p_n)$. Given a floorplan $\mathcal{F}$, power profile $\mathcal{P}$ and thermal properties of the

chip, the chip's maximum temperature, denoted by T_{max} can be determined by evaluating a function $\mathbf{T}$ over $\mathcal{F}$ and $\mathcal{P}$:

$$T_{max} = \mathbf{T}(\mathcal{P}, \mathcal{F}) \tag{1}$$

More details on thermal modeling can be found in the next section.

Your program should take as input $w_i, h_i, p_i, \quad i = 1, 2, \ldots, n$, and determines $\mathcal{F}$, such that T_{max} is minimized, under the constraint that:

$$\begin{gathered} b_i \text{ and } b_j \text{ do not overlap }, 1 \leq i < j \leq n \\ \frac{A_{chip} - A_{sum}}{A_{sum}} \leq \gamma \end{gathered} \tag{2}$$

where γ is a pre-defined threshold for area overhead.

3 Thermal Model

To estimate the chip's temperature (both transient and steady-state) based on a floorplan and a power profile, you should use the HotSpot thermal model proposed in [1]. HotSpot is a very widely-used thermal model based on an equivalent circuit of thermal resistances and capacitances that correspond to functional blocks and essential aspects of the thermal package [1]. More details can be found in [1].

For simplicity, **you only need to focus on steady-state temperature throughout this project**. You can view HotSpot as a black-box model: when given a floorplan $\mathcal{F}$ and a power profile $\mathcal{P}$, it will calculate the steady-state temperature of each block. Then the maximum stead-state temperature of all blocks is the T_{max} defined in Equation (1).

To use the HotSpot model, you can either embed HotSpot source code in your code, or make calls to HotSpot executable file. You should leave all the settings default as defined in the hotspot.config file when using HotSpot.

4 Input and Output Format

During the test, you will be given multiple test cases. Each test case has an input file "input.txt" that defines the dimension and power profile of functional blocks. The format of "input.txt" is as follows:

- The first line is an integer n, denoting the number of blocks in the circuit.
- The second line is a real number γ, denoting the allowed area overhead (see Equation (2)).
- Each of the next n lines defines a functional block. Each line has three numbers w, h, p, separated by space, where:
  - w is a real number, denoting the width of the block in meters.
  - h is a real number, denoting the height of the block in meters.
  - p is a real number, denoting the power of the block in watts.

Your should output the floorplan to file "yourUID.txt" (e.g. 112500000.txt) using the following format:

- There are n lines in the output file
- Each line defines the position of one functional block. It has two numbers x and y, separated by space, where

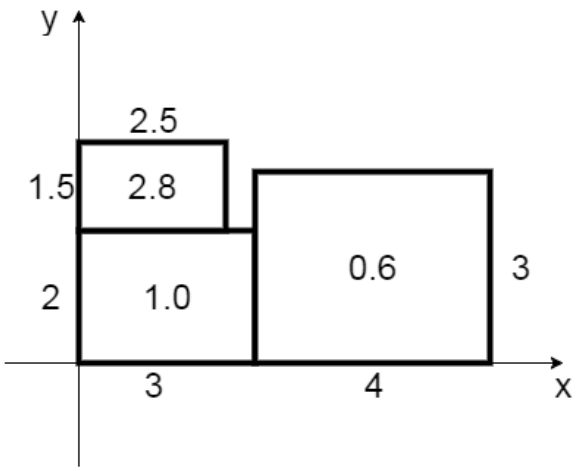

Figure 1: Sample floorplan. The dimension of each block is marked on the edge. The power of each block is marked inside the rectangle.

  - x is a real number, denoting the x coordinate of the bottom-left corner of the block
  - y is a real number, denoting the y coordinate of the bottom-left corner of the block

- The blocks should be in the same order as they are defined in input.txt file.

One input.txt example is:

```
3
0.25
3.0    2.0    1.0
4.0    3.0    0.6
2.5    1.5    2.8
```

One sample output corresponding to the above input file is:

```
0.0    0.0
3.0    0.0
0.0    2.0
```

The corresponding floorplan is shown in Figure 1:

Please strictly follow this input/output format and file naming conventions.

5 Grading

This project will be graded based on the optimality of your floorplan compared with other submissions. After submission, the TA will run your program on five input files. These input files will have exactly the same functional blocks (same dimensions and same power values) but different γ values (ranging from 0.2 to 0.8). The TA will calculate the maximum temperature using HotSpot corresponding to each floorplan your program generates. This gives a vector of five numbers, denoted by $\mathbf{T} = (T_1, T_2, T_3, T_4, T_5)$. Your program is said to be optimal if there does not exist another $\mathbf{T'} = (T'_1, T'_2, T'_3, T'_4, T'_5)$ such that $T'_i \leq T_i$ for $i = 1 \ldots 5$ and at least one inequality is strict. Optimal programs will receive 100 pts while other sub-optimal programs will receive points based on how suboptimal they are.

Please note that **the run-time limit of your program is 10 minutes**. If after 10 minutes your program does not finish, the TA will kill your program. If there is an output file generated, the TA will use that file to calculate the maximum temperature. If not, then the maximum

temperature corresponding to that input will be set to 600K (which is a unrealistcally high temperature value), as a penalty.

6 Deliverables

Please submit the following materials:

- **Source code** of your program. Please include a README file describing how to compile and run your program. Note that **your program has to be compilable and executable on the glue system**.
- **A report** briefly explaining how your floorplanner works including algorithms and data structures. Please include figures if necessary. The report is limited to 5 pages maximum.

7 Attachments

The following files have been attached:

- *input.txt* : a sample input file. You may evaluate the performance of your program on this file. However, the input file used during the grading will differ from this one.
- *hotspot.zip* : Source code of HotSpot, version 5.02.

References

[1] Wei Huang, Shougata Ghosh, Siva Velusamy, Karthik Sankaranarayanan, Kevin Skadron, and Mircea R Stan. Hotspot: A compact thermal modeling methodology for early-stage vlsi design. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(5):501–513, 2006.