Simple Simulated-Annealing Cell Placement Tool

Digital Design II

Omar Fayed 900191831

Youssef Abouelenin 900192509

Table of Content

Objectives

- Develop a simple simulated annealing-based placer that minimizes the total wirelength.
- Understand the effect of cooling rate on the quality of the placement.
- Enhance the half-perimeter wire length by obtaining the smallest bounding box for every net.

Introduction

Cell placement is a very important step in the production of integrated circuits. It is the third step in the physical design procedure, following floor-planning and power planning. It is very significant because it impacts the quality of routing. Improving the placement of cells minimizes the total wire length, decreases the routing congestion, and decreases the overall delay. In this project, we will improve the placement by using a simulated annealing placer tool. The tool does not provide an optimal solution, but it is good at avoiding local minima.

Algorithm

In the simulated annealing algorithm, we first perform an initial random placement of the cells. Then we set the temperature at a very high value, so we can avoid being trapped int a local minima. The temperature is responsible for the rejection probability of swapping two random cells. After setting the temperature, we start looping until the temperature goes lower than the final temperature. In the loop, we pick two random cells and swap them in each iteration. Then, we calculate the change in the total wire length. In our case, we used the half-perimeter wire length (HPWL) estimation to obtain the total wire length in our grid. If the wire length decreases, we accept this swap and proceed; otherwise, we reject the swap with probability (1 - e^(delta(L)/T)) and we proceed in looping.

The cooling schedule is the method which reduces the temperature parameter. If the temperature parameter drops quickly, this will have a big impact on the quality of the solution. Also, dropping the temperature slowly would result in a longer computing time, thus the cooling schedule is vital for the quality of the solution.

The following is the cooling schedule used in this project:

- Initial Temperature= 500 x Initial Cost
- Final Temperature= 5x10-6 x (Initial Cost) / (Number of Nets)
- Next Temperature= 0.95 x Current Temperature
- Moves/Temperature= 10 x (Number of cells)

Functions

1. Hpwl(): used to obtain the half-perimeter wire length of the connection by using minimum and maximum values of rows and columns.
2. Swap(): used to swap the values of two cells
3. Anneal(): initially sets the cells randomly, then finds the wire length with the least cost.
4. MaxHpwl(): finds the maximum hpwl by comparing the wire length of each connection in a net.
5. Print(): prints the grid