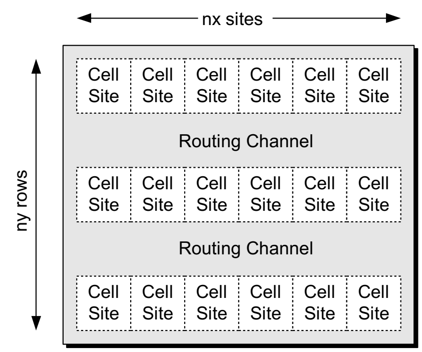
UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CPEN 513: CAD Algorithms for Integrated Circuits
2022/2023 Term 2

Assignment 2
Due: Monday February 20th by 11:59pm

In this assignment, you are to implement a simulated-annealing based placement tool that targets standard cells. Your tool is to assign physical locations to each cell in a circuit. The cells are to be laid out in rows as shown below:

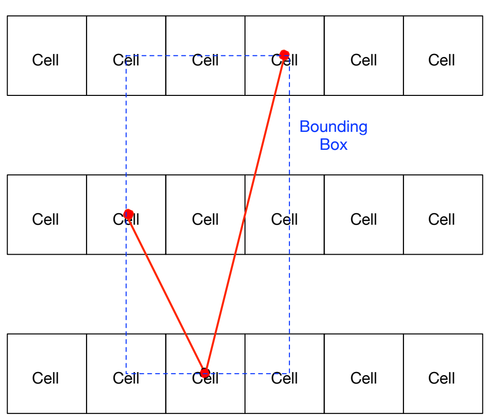


Each cell is square and is the same size. The final placed circuit will contain *ny* rows, each with *nx* cells (the value of *nx* and *ny* depend on the size of the user circuit and will be given to you in the input file: see below). A valid placement is one in which no site is assigned more than one cell. Note that, at this stage of the CAD flow, you are not worried about routing or adding feedthroughs; all you care about here is assigning physical locations to all cells in the circuit.

The optimization goal is to minimize the half-perimeter of the smallest bounding box containing all pins for each net, summed over all nets (this is the cost function we talked about in class). You can assume the following:

a) The routing channel is the same height as one row of cells (this is not necessarily realistic in today's technologies, but it is what we will assume).
b) The distance between two cells can be measured from the center of one cell to the center of the other. Don't worry about whether the cell pin will be on the top or bottom of a cell; this is for the router to worry about.
c) There is no extra, artificial, penalty for a net which crosses a row.

As example, the three-terminal net shown in red in the following diagram would contribute 6 to the cost function (width of the bounding box (blue) in the x dimension is 2, and height of the bounding box in the y dimension is 4 – remember to include the space between the rows).

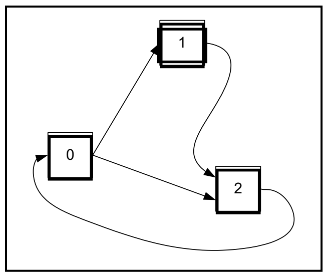


The circuit input format is as follows. The first line contains the number of cells to be placed, the number of connections between the cells, and the number of rows and columns upon which the circuit should be placed. Note that the product of the x and y dimensions should be at least as large as the number of cells in the circuit. In the following example, there are 3 cells, 4 connections between the cells, and the circuit should be placed on a chip with two rows of two cells each. `3 nets (typo)`

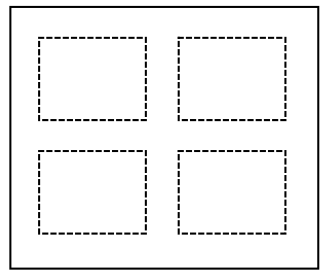
The netlist file then contains one line per net. Each net can connect to two or more logic blocks. For each line, the first number is the number of logic blocks to which this net connects. The remaining numbers are the block numbers connected to this net. Note that blocks are numbered from 0. So, in the following example, there are three nets: (1) one that connects blocks 0, 1 and 2, (2) one that connects block 2 to block 0, (3) one that connects block 1 to block 2.

```
3 3 2 2
3 0 1 2
2 2 0
2 1 2
```

example netlist



circuit corresponding to example netlist



target chip on which this circuit should be placed

As in Assignment 1, your program should employ graphics to show the progress of the algorithm as it proceeds. Again, it is not enough just to show the final answer. After a placement, you should report the sum of the half-perimeters for each net (this is your cost function).

Test your circuit on each of the files on the Canvas Assignment 2 directory (a .zip file and a .tar file are provided; use whichever you prefer). For each circuit, you should report the final cost function, and average overall all benchmark circuits. (hint: if you get an overall average around 2000-3000, that is pretty good. If it is more than 4000, see if you can do some optimization). If you get more than 10,000, then you are probably “quenching” (ie. lowering the temperature too fast or not choosing suitable start and end temperatures).

What to hand in:
You must handin your files using the Canvas site. You should hand in a zip file containing the following:

- A text of PDF file (report.pdf or report.txt) containing:
  - A description of how your program works (assuming I already have a basic idea of how simulated annealing works). Be sure to explicitly address each of the criteria in the attached marking scheme. In particular, you should include your annealing schedule, as well as anything you did to ensure the program runs fast. If you are hoping for “initiative” marks, you need to justify them in your report.
  - A table of results for the example files
  - One or more figures showing the graphics

- The source code. You can either provide a link to an on-line repository or include the source code in your Zip file.

Marking Scheme for Assignment 2: Maximum score = 24

Basic Simulated Annealing Algorithm

Score	0	2	4
Description	The implementation does not perform simulated annealing correctly.	The implementation performs simulated annealing, but problems evaluating moves properly.	The implementation correctly performs simulated annealing.

Cost Function Calculation:

Score	0	1	2
Description	The cost function calculation is not correct.	The cost function calculation is not correct, but is within a constant factor.	The cost function calculation is correct.

Efficiency:

Score	0	2	4
Description	The implementation is not efficient; run-time is much longer than it needs to be.	The implementation is somewhat efficient, but small changes could make it faster. (example: cost is calculated from scratch for each move)	The implementation is efficient; there is clear evidence of care creating an algorithm that runs fast.

Results (Annealing schedule tuning): How good are the results (compared to the rest of the class?).

Score	0	2	4
Description	The results (final cost function) is among the worst 25% in the class.	The results (final cost function) is better than the bottom 25% of the class, but worse than the top 25% of the class. The report contains evidence that some attempt at schedule "tuning" was made.	The results are among the best 25% of the class. The report contains good evidence of schedule "tuning"

Graphics:

Score	0	1	2
Description	No graphics.	The graphics only show the final solution (or a small number of intermediate solutions)	The graphics clearly shows the progress of the algorithm as it progresses.

Code Quality:

Score	0	1	2	3
Description	Code is lacking in structure and comments.	Code is of sufficient "academic code" quality, including some comments.	Code is of good "academic code" quality including extensive comments. Code is well structured.	Code is of industrial quality, including evidence of unit tests and/ or extensive system tests.

Report (maximum 3 pages):

Score	1	2	3
Description	Report is unclear or difficult to read and/or understand..	Report describes most aspects of marking scheme clearly. The English has grammar/clarity errors that would not be acceptable in a major IEEE or ACM journal or conference.	Report describes all aspects of marking scheme clearly. The English is of a professional standard that would be acceptable in a major IEEE or ACM journal or conference.

Initiative: (be sure to identify any extensions in your report):

Score	0	1	2
Description	Assignment implemented as in handout.	The implementation contains one or more straightforward extensions.	The implementation goes beyond what is described in the handout in a non-trivial way.