

ECE 556 Design Automation of Digital Systems (Spring '24)
Programming Assignment 3 - Report
Noah James Degroot, Dan Jose Jinoy, Adithya Pillai Ramesh

I. Compiling and Running the Code

The code is split into two files pa2.h and pa2.cpp. Both these files can be compiled together using the command

g++ -O3 -o [executable_name] pa3.h pa3.cpp

The -O3 argument performs compiler optimisations for memory/compile time intensive operations.

The code can be executed using the following command.

./[executable] [input_block_file] [output_file]

II. Routing Results

Input File	Runtime(ms)	Wirelength
case1	0	337
case100000	32	2497266886718
case2	0	1346
case200000	63	5010565565386
case3	1	2755
case4	1	125026
case5	1	24930123
case500000	128	125105012
case6	2	497848917
case7	2	12558443485
case8	4	250924413178

III. Challenges Faced

The input file consisted of the Boundary, number of pins and the position of each pin. Processing in the data from the file involved reading the lines and looking for non essential elements like parentheses and commas and removing them before assigning the values to their data structures.

The Steiner tree implementation for routing typically follows the Minimum spanning tree algorithm. We tried implementing the minimum spanning tree algorithm but were not

successful so we came up with a simple and straightforward approach to building a steiner tree. Using the boundaries as a bounding box, we assumed a line which horizontally cut the bounding box into equal halves. Using this line as a Steiner tree, we routed the pins through this horizontal line and thus built the Steiner tree, meaning the points at which the routes from the pins cut the line are the Steiner points. This idea for building the Steiner tree can be scaled by adding vertical lines and routing via the horizontal or the vertical line using a simple algorithm.