EECE 583 Assignment 1: Routing

Ken Mansfield

January 25, 2016

Setup

I have chosen to write the assignment using C++ so that I could utilize the provided easyGL graphics package to visualize the results. Using easyGL was not as simple as I had hoped since it did not work on a fresh install of Ubuntu, I spent a couple hours manually installing several packages and hundreds of fonts until it finally worked.

Input

The first thing the code does is read the two command arguments – file name and mode. The mode is used to determine how the program will be visualized and the file name is the netfile that will be read. The file is opened and each line read individually and parsed. The details are all saved into a NetFileStruct that specifies each item.

Attempting a Layout

Failures to place a wire are usually due to another wire blocking it off. This can often be fixed by simply changing the order in which wires are placed. The code first attempts 1 order, and if it fails it changes the order. It does this using std::next_premutation(prem.begin(), prem.end()) which re-orders the vector on each iteration until it tires all premutations and returns false, in which the code ends the loop. Trying every permutation is an O(n^2) operation which is easily possible with our sample files since the number of wires is fairly small. Failing to route a layout with 8 wires only took 40,320 iterations (less than 10 seconds).

Shortest-Distance Algorithm

CreateBlankCircuit(…) creates and draws 2D grid, then BuildCircuit(…) places and draws all the cells within the circuit so that nets will be blocked off from those squares. routeWire(…) is where the routing algorithm takes place. Following the instructions in the lecture notes, the code starts by pushing the starting grid point into an expansionList vector. Each neighbor is given an incremented value and pushed into the vector (and the previous grid point is removed). The loop continues until if finds all outputs. If it fails to find the outputs, the expansion list will eventually become empty when there are no more grid points to add the list, then the function will return false notifying the program that routing failed.

Once values have added to each individual grid up until the output sinks, we can retrace the shortest distance back to the input. Since the goal of this algorithm is to simply find the shortest distance for each input/output pair, we can simply just re-trace the route for each output individually. This method does not

reduce the total amount of wire placed, but it does find the shortest distance between each input/output pair.

Testing

Beyond using many cout statements and debugging using GDB/DDD, I implemented a test script that runs through every .infile in the current directory. Large-scale software projects always have automated testing scripts that run at specified intervals (such as whenever a file is updated in the repository) to ensure that there are no unexpected results. The testAllFiles.sh will loop through every file and run the routing program on each one in command line only mode.

Graphics

To visualize the progress of the algorithm (which is also very useful for debugging) the easyGL package was used. First a grid is displayed which takes the X,Y grid sizes to determine the size of each grid cell. For each cell, a blue square is drawn. A number is also placed where each of the wire inputs/outputs are located. During the execution of the shortest-distance algorithm the current cell value is displayed so that we can track the progress of its search for each output. Once the output is found, the net is drawn with its own unique color.

Modes

Visualizing the progress in slow-mode is fun and useful at first but it takes a long time to get through larger circuits. I have added 4 different modes:

0 – Normal Mode

1 – Faster mode. The time step between updating the drawing is made 5x faster.

2 – No animations. Only the final result is shown once the nets have been placed.

3 – No graphics. This is text only mode which is useful for running automated testing.

Usage

./assignment1.exe [input file] [mode]

Ie.

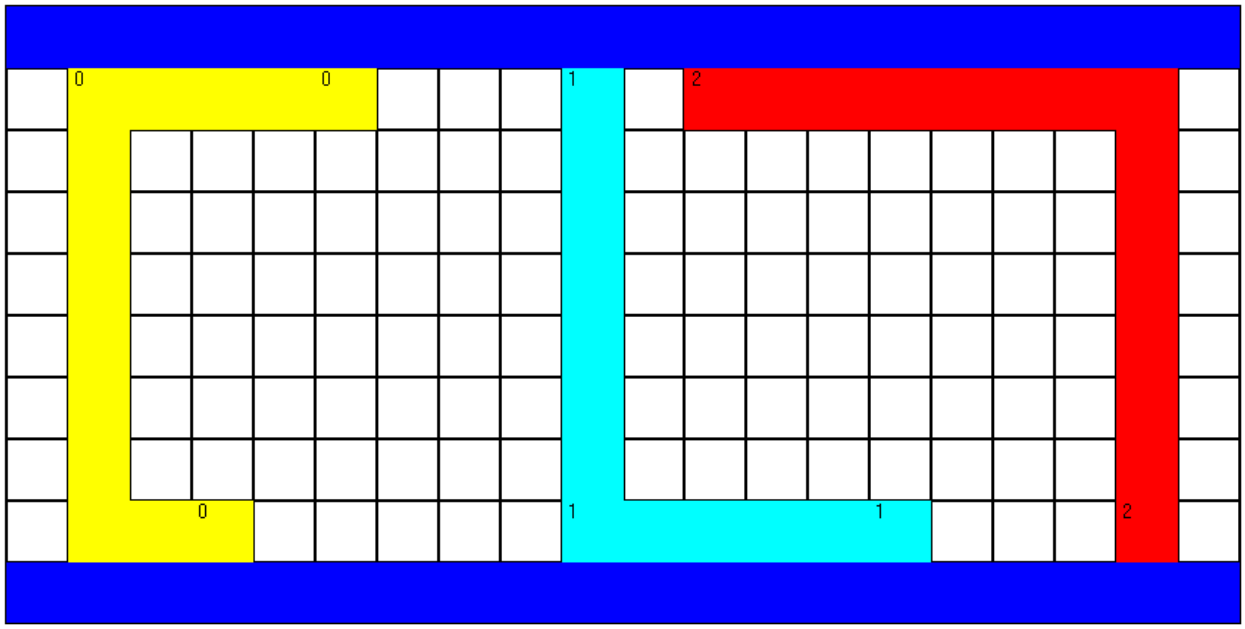
./assignment1.exe stanley.infile 0

Or the test script can be run:

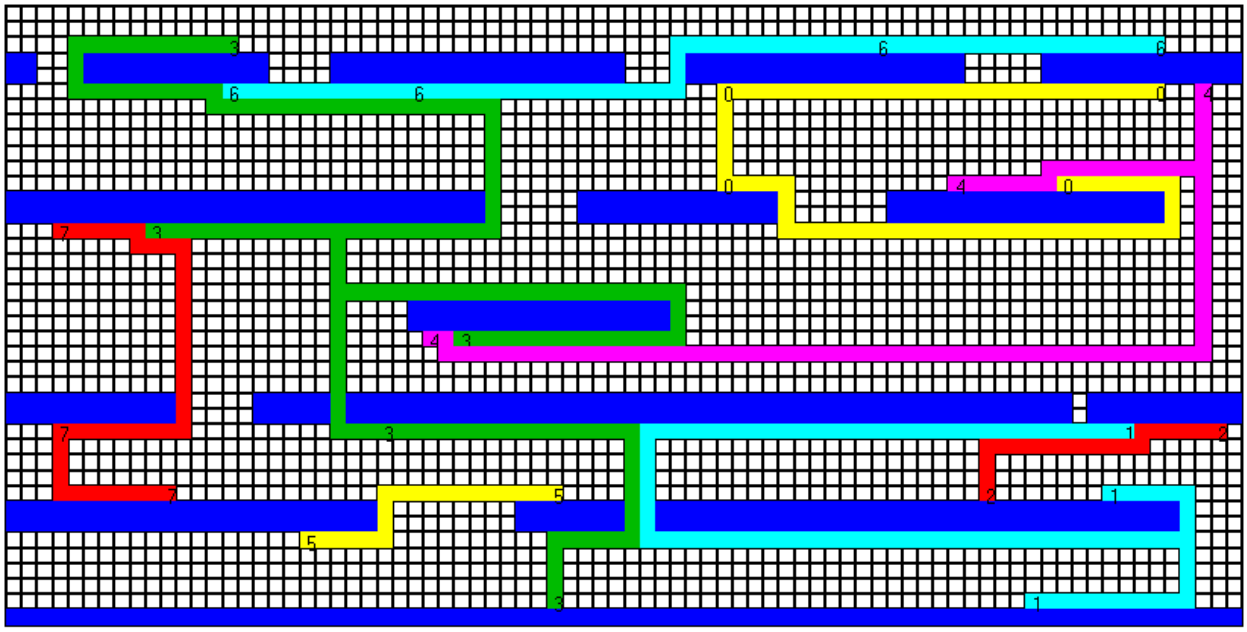
./testAllFiles.sh

Results

stanley.infile



stdcell.infile



Wires successfully placed per netfile. “All” means that it successfully placed all wires.

File	Wires Placed
impossible2	2
impossible	2
kuma	3
misty	all
oswald	1
stanley	all
stdcell	all
sydney	all
temp	5
wavy	all

```
Processing impossible2.infile file..
./assignment1.exe impossible2.infile 3
*** Unable to route this circuit after 1 premutations ***
This particular iteration failed on wire: 2
Successfully wired: 2

Processing impossible.infile file..
./assignment1.exe impossible.infile 3
*** Unable to route this circuit after 6 premutations ***
This particular iteration failed on wire: 2
Successfully wired: 1

Processing kuma.infile file..
./assignment1.exe kuma.infile 3
*** Unable to route this circuit after 15 premutations ***
This particular iteration failed on wire: 1
Successfully wired: 3

Processing misty.infile file..
./assignment1.exe misty.infile 3
Attempted nets in this order: 3 2 1 0
*** Successfully routed all nets ***

Processing oswald.infile file..
./assignment1.exe oswald.infile 3
*** Unable to route this circuit after 2 premutations ***
This particular iteration failed on wire: 1
Successfully wired: 1

Processing rusty.infile file..
./assignment1.exe rusty.infile 3
Attempted nets in this order: 2 1 0
*** Successfully routed all nets ***

Processing stanley.infile file..
./assignment1.exe stanley.infile 3
Attempted nets in this order: 2 1 0
*** Successfully routed all nets ***

Processing stdcell.infile file..
./assignment1.exe stdcell.infile 3
Attempted nets in this order: 7 6 5 4 3 2 1 0
*** Successfully routed all nets ***

Processing sydney.infile file..
./assignment1.exe sydney.infile 3
Attempted nets in this order: 2 1 0
*** Successfully routed all nets ***

Processing temp.infile file..
./assignment1.exe temp.infile 3
*** Unable to route this circuit after 31 premutations ***
This particular iteration failed on wire: 1
Successfully wired: 5

Processing wavy.infile file..
./assignment1.exe wavy.infile 3
Attempted nets in this order: 0
*** Successfully routed all nets ***
```