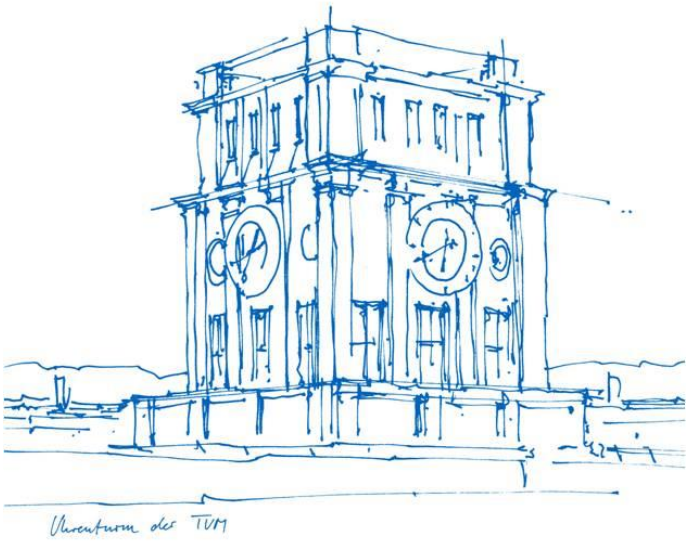
INTERNSHIP REPORT

(Jan. 11, 2019 – April 18, 2019)

Design Interpreter Development for Microfluidic Simulation



Advisor: Dr.-Ing. Tsun-Ming Tseng

Supervisor: Prof. Dr.-Ing. Ulf Schlichtmann

Chair of Electronic Design Automation

Technical University Munich

Submitted By

Yanlu Ma

Date of Submission April 23, 2019

Table of Contents

Acknowledgement

First of all, I wish to express my deep gratitude to Prof. Dr.-Ing. Ulf Schlichtmann, Head of Chair Electronic Design Automation, and to Dr.-Ing. Tsun-Ming Tseng, Research and teaching assistant at the Chair of Electronic Design Automation, for giving me the opportunity to do this internship.

I sincerely thank Ms. Mengchu Li for her guidance and encouragement in carrying out this work. Without her kind direction and proper guidance this internship could not have been successfully conducted. I choose this moment to acknowledge her contribution gratefully.

I am also grateful to my friends who helped me while preparing the study by giving their suggestions and supply of information, which were valuable to me.

1. Introduction

1.1 Background

With the development of biology the observation of interactions between DNA, RNA and proteins is getting more important. In order to provide the technical support, "microfluidic " systems have potential ability in a diverse array of biological applications, especially in fabrication of biochips. Current design automation research for continuous-flow microfluidics deals based on multi-layered valve and pump technology, which is fabricated by soft lithography (Todd Thorsen, 2002). The advantages of Soft lithography are biocompatibility and ease of fabrication. Continuous-flow microfluidic large-scale integration (mLSI) has its advantages in high throughput and in accurate control of reagent volume (Tsun-Ming Tseng, 2017). As shown in Figure 1, the mLSI chip has small size but with sufficient functionality.

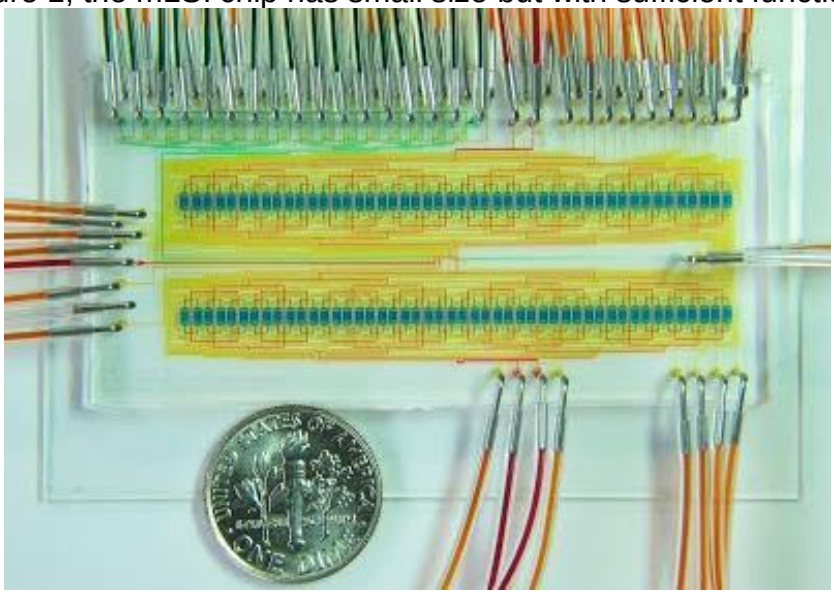


Figure 1: Standard mLSI chip. The device shown above is a 96-well cell culturing system that operates through a microfluidic platform (Schmidt & Fitzgerald, 2017)

To understand the operation of mLSI devices, it is necessary to understand the main structure. In mLSI, the main structure consists of two layers, a control layer and a flow layer. On the silicon base the Polydimethylsiloxane (PDMS) material is used. (Marc A. Unger, 2000) Each layer has several channels, which are connected with different external sources. Fluids like reaction sample and reagents can be imported from or exported to external source. The layer interaction happens at valves. Valves are designed to guide the fluid direction in the flow layer. The valve actuation relies on the change of pressure in the control layer. (Marc A. Unger, 2000) The flexible membrane can be pressed from control layer to flow layer and the state of valve changes from open to closed (Tsun-Ming Tseng, 2017). State: open indicates valve is not actuated and state: closed indicates valve is actuated. With help of valves the accurate control of reagent volume can be fulfilled. Because of the interaction between control layer and flow layer, the design of two layers is highly dependent.

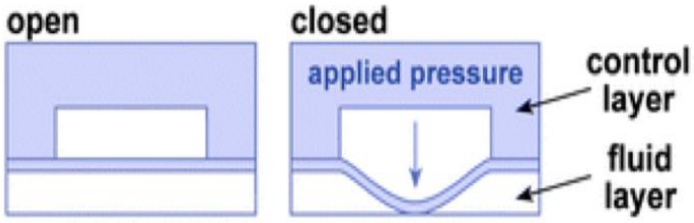


Figure 2: Side view of a push down before and after pressure application. (Schmidt & Fitzgerald, 2017)

1.2 Columba

In recent years, engineers are trying to implement design automation replacing the manual design. However, the previous designs of mLSI just simplify the interaction between two layers. It is common that two flow channels cross in coin-sized area. And corresponding valves are used to guide the fluid direction at flow channel crossings. These additional valves result in additional control channels. It is essential to synchronize the two layers whenever flow channels crosses. (Tsun-Ming Tseng, 2017) In order to model the microfluid system accurately, a co-layout synthesis tool named Columba is created. Columba is the first design automation tool for mLSI. It takes a plain-text netlist description as its input and outputs an AutoCAD-compatible skript.

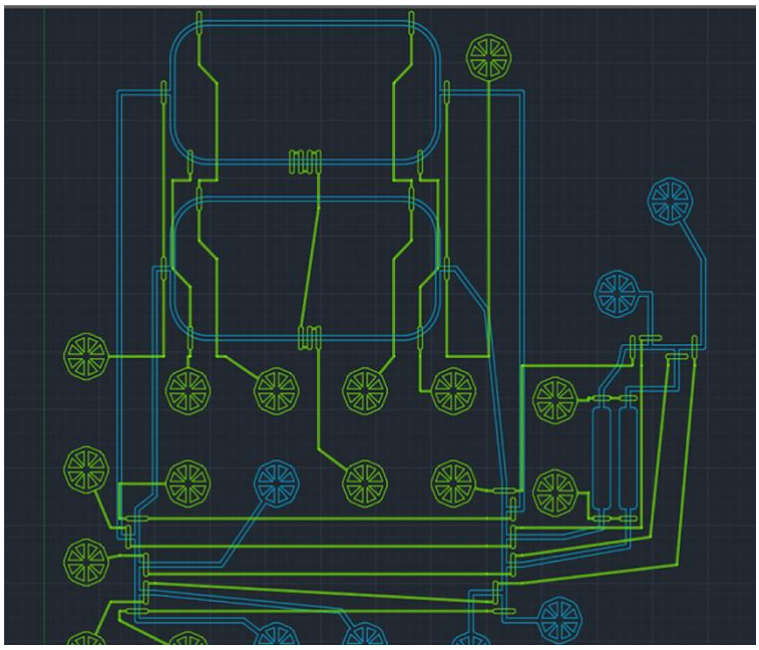


Figure 3: Generated by Columba

MODULE MODEL FOR MICROFLUIDICS

In addition to two layers, the initial physical architecture for a specific type of microfluidics can be implemented in different module models. Columba uses pin binding options to actuate valves (Tsun-Ming Tseng, 2017). The selection of pins determines the position of valves and the corresponding connection of control channels. In the following figures, the control channels are drawn as green lines, the flow channels are drawn as blue lines and valves are drawn as orange rectangles. Two definitions have to be introduced, before we talk about the module models. They are branch and edge. If two flow channel crosses, the channel junction is called branch. The paths between branch and valve, valve and valve, inlet and valve are called edge.

- Inlets/outlets: Inlets/outlets are used to connect with external source. Due to the limitation of inlets/outlets on a chip, several valves have to be connected together.
- Mixer: Mixer consists of two in-out flow channels, a ring-shaped flow channel, six position fixed valves, two pumps and several control channels. Only one of these pumps needs to be applied in final design, the other one will be removed. (Tsun-Ming Tseng, 2017) In Columba 2.0 the number of valves within pump should be chosen at least three. This means, the total number of valves from one mixer is nine or ten. Besides, mixer has 25 pins to actuate valves and two branches. Due to the unsymmetrical structure, mixer has four orientations, two for horizontal and two for vertical.
- Reaction chamber: A reaction chamber consists of two in-out channels and one flow channel with adjustable width and length. It has two valves for each in-out channel. Chamber has no branch in its structure.
- Switch: Switch is used to guide the fluid direction when flow channels cross. (Tsun-Ming Tseng, 2017) It has one main flow channel and several channel junctions. As shown in Figure 5, the number of channel junctions indicates the number of branches in the switch. For each branch, only one of the opposite flow channels will be chosen in the final Columba design.

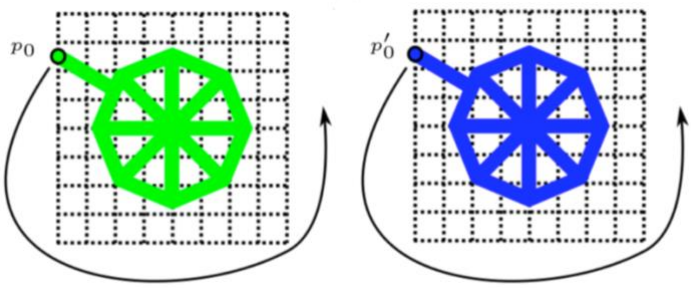


Figure 4:: Module models: Inlets/Outlets of control channel(left) and flow channel(right) (Tsun-Ming Tseng, 2017)

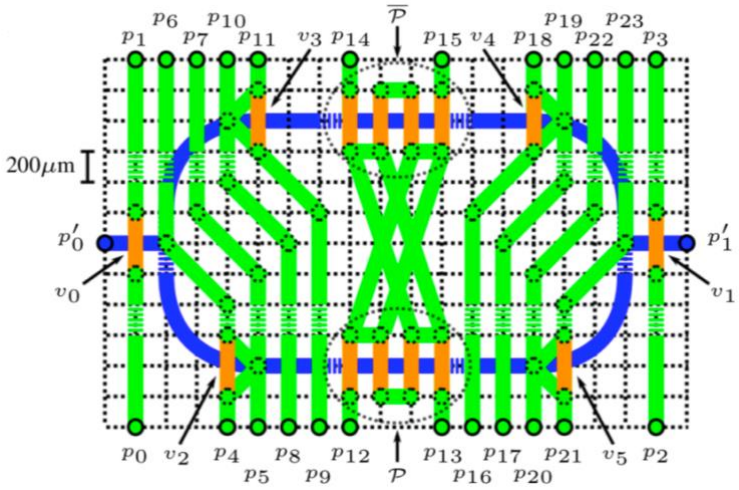


Figure 5: Module model: Mixer (Tsun-Ming Tseng, 2017)

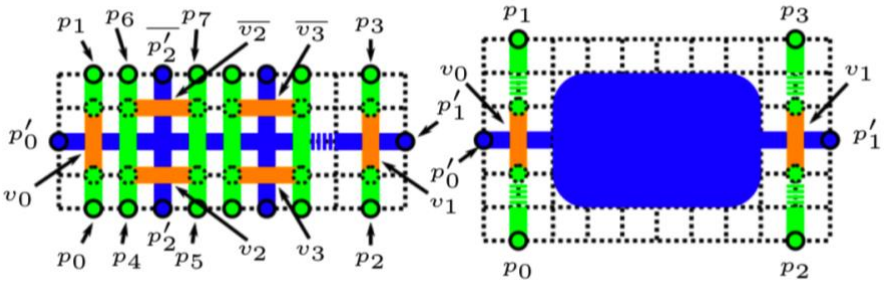


Figure 6:Module models: Switch(left) and Reaction chamber(right) (Tsun-Ming Tseng, 2017)

2 Description of Internship

2.1 Motivation

Current research has automatically generated the physical design, operation scheduling and binding protocol. This automatic synthesis has a missing link between the design and protocol. There is one new research taking the design and protocol as inputs, aiming to validate the flow-path and control-sequence. After simulation, the results can be imported into Columba as input and Columba can generate a new physical design with better performance, such as execution runtime reduction. But the output of Columba, i.e., the AutoCAD-compatible script using a series of drawing command, cannot be directly used as the input for the new research.

2.2 Input and Outputs

This work takes the Columba output as the input. The output of Columba is a text file that only contains types of module models, coordinates and pin allocations. What I have to do in this Internship is to develop a tool to automatically mark out all the “landmarks” including inlets/outlets, valves, branches and edges; to calculate the length and width of edges; and to mark all the valves which are connected to same control pin. The output of my work should also be a text file. Table 1 shows an example output.

TABLE I: Generated output design

In-/Outlets: 1 2 3 4 5 6 7 end Branch: 8 9 10 11 12 13 14 15 16 17 18 19 20 end Valve: 21 22 23 … 57 58 59 end Edge: 8 9 13.33 3.33 …… 1 24 41.03 3.33 end CChannel: 1 21 26 …… 15 56 58 end

The output text file contains five different categories, including In-/Outlets, Branch, Valve, Edge and CChannel. All categories end with a word "end". The numbering of first three categories is natural number starting from 1. The first two columns of category Edge are number of valves or branches or inlets, the third column is length of corresponding flow channel and the fourth column is width of corresponding flow channel. The last category contains all the number of valves, which are connected with same control pins. Moreover, visualization of numbering is essential. As shown in Figure 7, all the numbers are visualized.

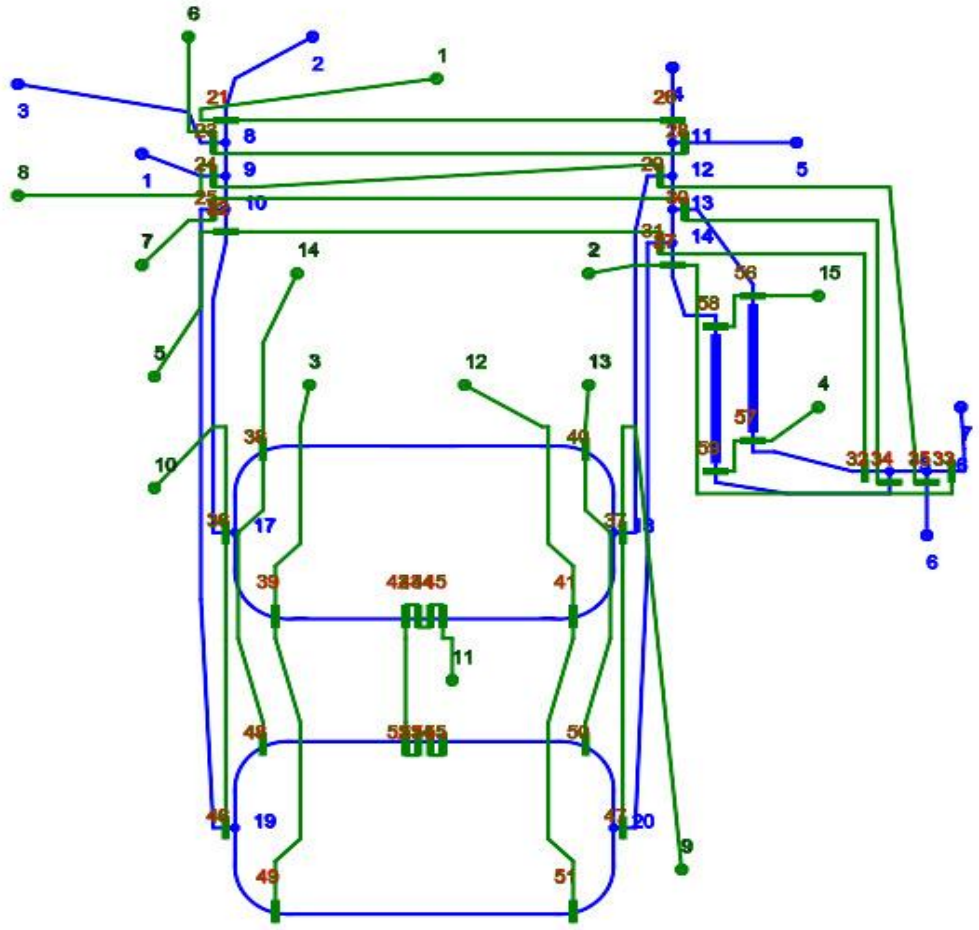


Figure 7: Example of Visualization and numbering

3.Implementation

3.1 Design

- Understand the construction of module models.
  The output of Columba consists of only the coordinates of center point and the reference coordinates for determination the feature points, the size of whole design stored in variable maxX and maxY, the type of devices and the orientation of modules.
  - ➢ Mixer: 12 feature points. Paths between 1-2 and 3-4 are in-out channels. Paths between 5-12, 6-7, 8-9,10-11 are straight lines. Paths between 5-6, 7-8, 9-10, 11-12 are ¼ circle with radius 283.3 µm.
  - ➢ Switch: 2+2*n feature points. n represents the number of branches(purple). Path between 1-2 is the main flow channel. Lines between two circles are valves.
  - ➢ Reaction Chamber: 4 feature points. Paths between 1-3 and 4-2 are in-out channels of chamber. Flow channel between 3-4 can be described as rectangle.
  - ➢ Inlets/Outlets of I: They are simplified as one single point.
  - ➢ Channels: Coordinates of starting points and destination points.
- Implement in C++ and get the coordinates of feature points
- Numbering all the inlets/outlets, branches, valves
- Matching the module models with channels and determine edges.
- Calculating the edge length and width

Figure 8: Feature points of Mixer

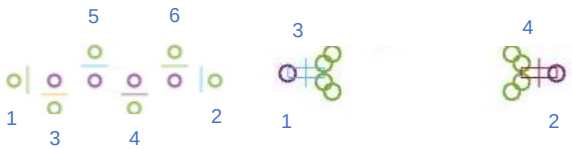


Figure 9: Feature points of Switch(left) and Reaction Chamber(right)

TABLE II: features of all module models

	# of type	# of Feature points	orientation	Pin allocation	# of valves	# of branches
Mixer	1	12	4	25	6 + X	2
Switch	4	2 + 2*n	2	8	2 + n	n
Chamber	2	4	2	4	2	0
Control channel	-2	2	-	-	-	-
Flow channel	-1	2	-	-	-	-
Control inlets	5	2	-	-	-	-
Flow inlets	3	2	-	-	-	-

X depends on the pin allocation of mixer. It could be 3 or 4.

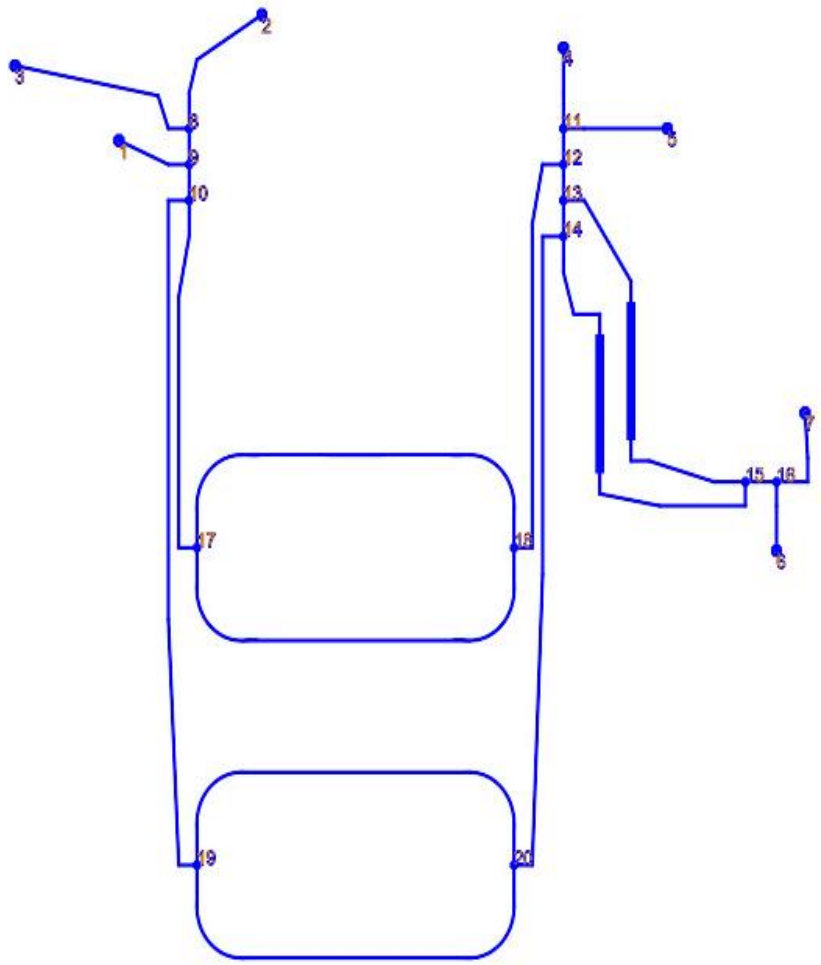


Figure 10: Example of flow layer

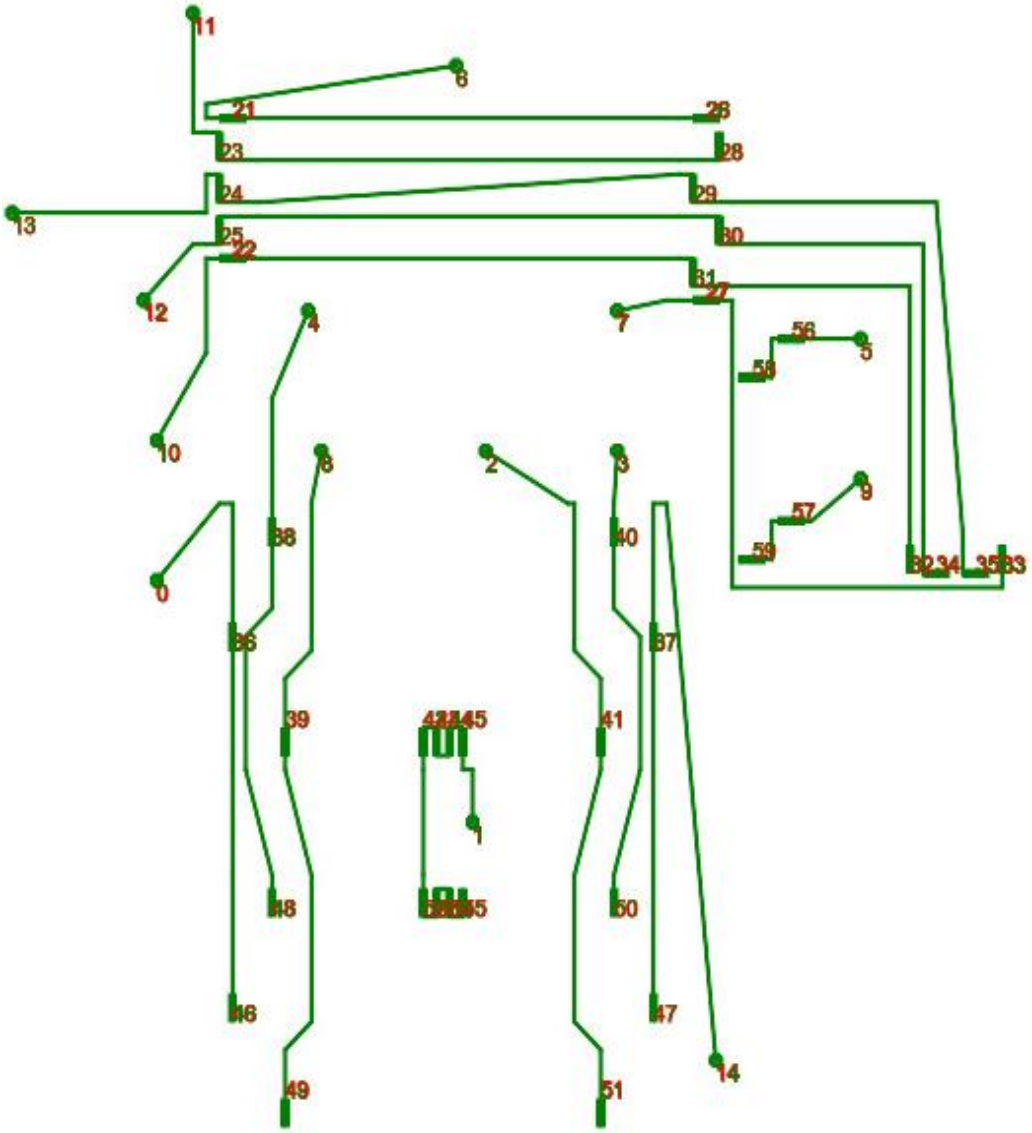


Figure 11:Example of Control layer

3.2 Visualization

This work is implemented in C++ using Xcode. Xcode console cannot show graphics directly. In order to visualize the design, I use Imagemagick to create the image in form of JPEG. First of all, the Magick++ library has to be linked in Xcode. (Magick++, n.d.)

Step 1: Install Imagemagick with homebrew in Terminal

Command: *brew install imagemagick*

Step 2: Compile and link all packages

Command: *pkg-config --list-all | grep -i Magick*

Step 3: Get the Header File Path

Command: *pkg-config --cflags-only-I Magick++*

Step 4: Get the Search Patch

Command: *pkg-config --libs magick++*

Step 5: Get all flags for linker

Command: *Magick++-config --cppflags --cxxflags --ldflags –libs*

Step 6: Copy all the corresponding results to Build Settings in Xcode

Step 7: Check the configuration by initializing the ImageMagick library

In Xcode using built in Function: InitializeMagick(); (Setchell, 2016)

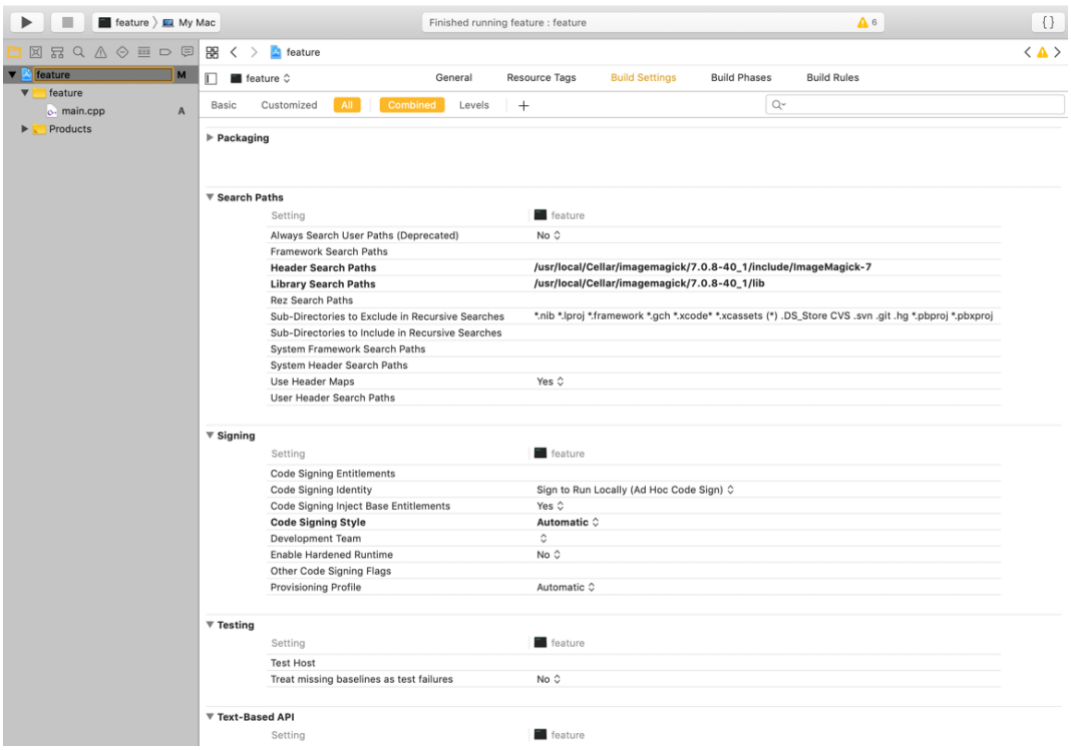


Figure 12: Header and Library Search Path in Xcode

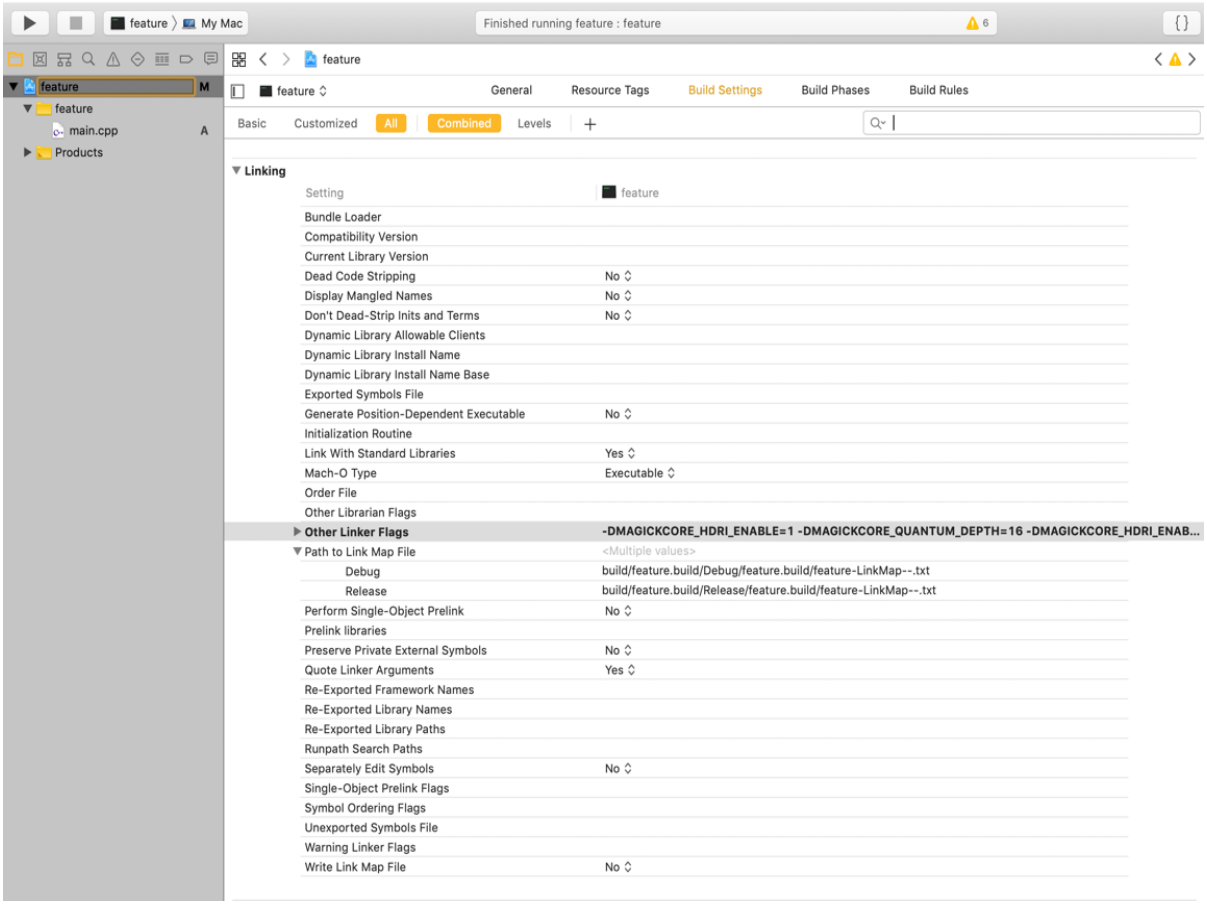


Figure 13: Linker Flags in Xcode

After Installation and configuration, using built-in function drawableLine(), drawableCircle() and drawableRectangle() to connect the feature points.

4. Experimental Results

This program is implemented in C++ and I test its performance with two test cases. These two test cases are generated by Columba. Flow channels are drawn in green lines. Control channels are drawn in blue lines. In order to show the numbering clearly, the colors of number are different. The numbering of flow inlets/outlets is in green. The numbering of control inlets/outlets are in blue. The rest are in red. Comparing with images generated by AutoCAD, the numbering of program is correct and matches perfectly. The width of edge between two chamber valves is significant bigger than the flow channel width.

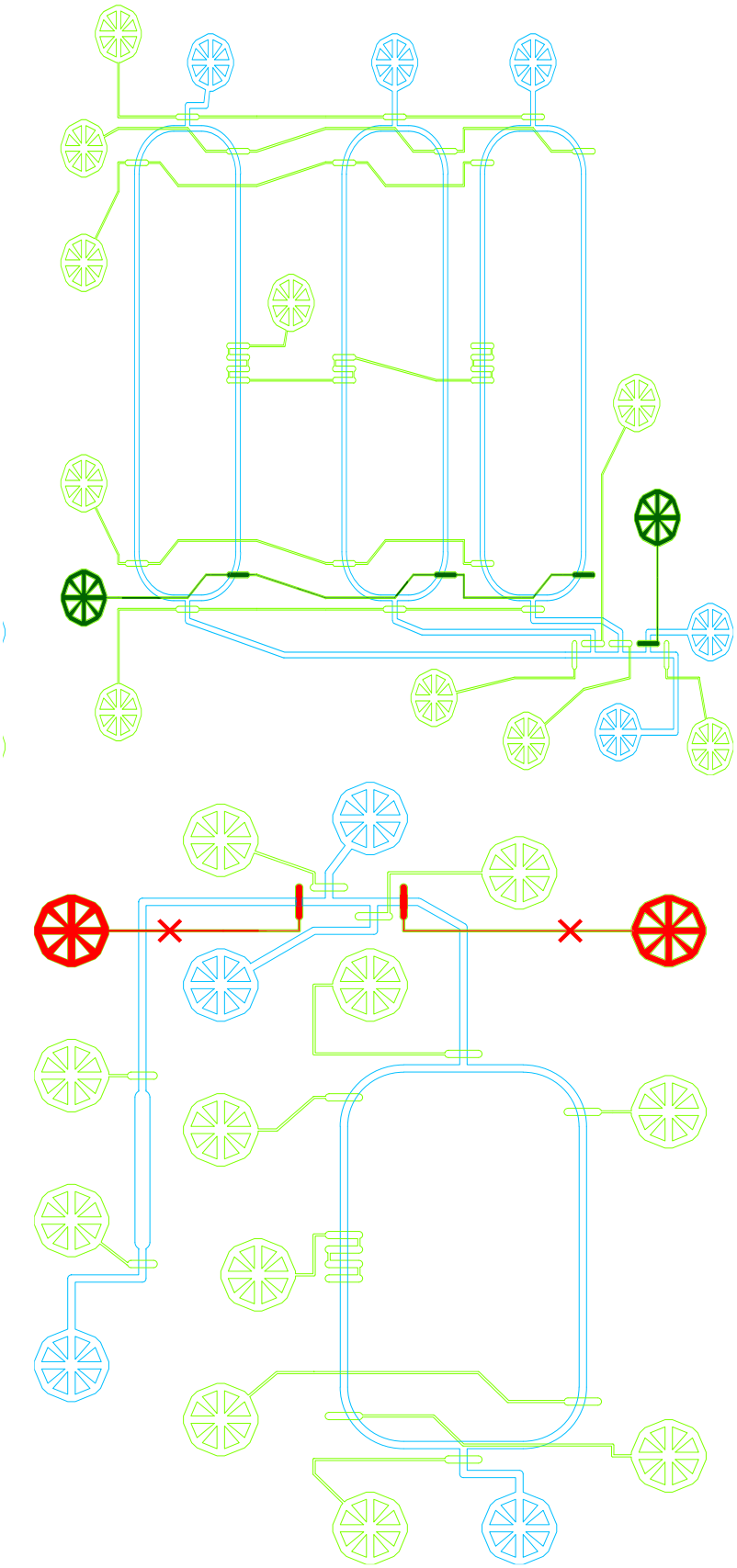


Figure 14: Testcase generated by Columba. Testcase 1 (bottom) Testcase 2(top)

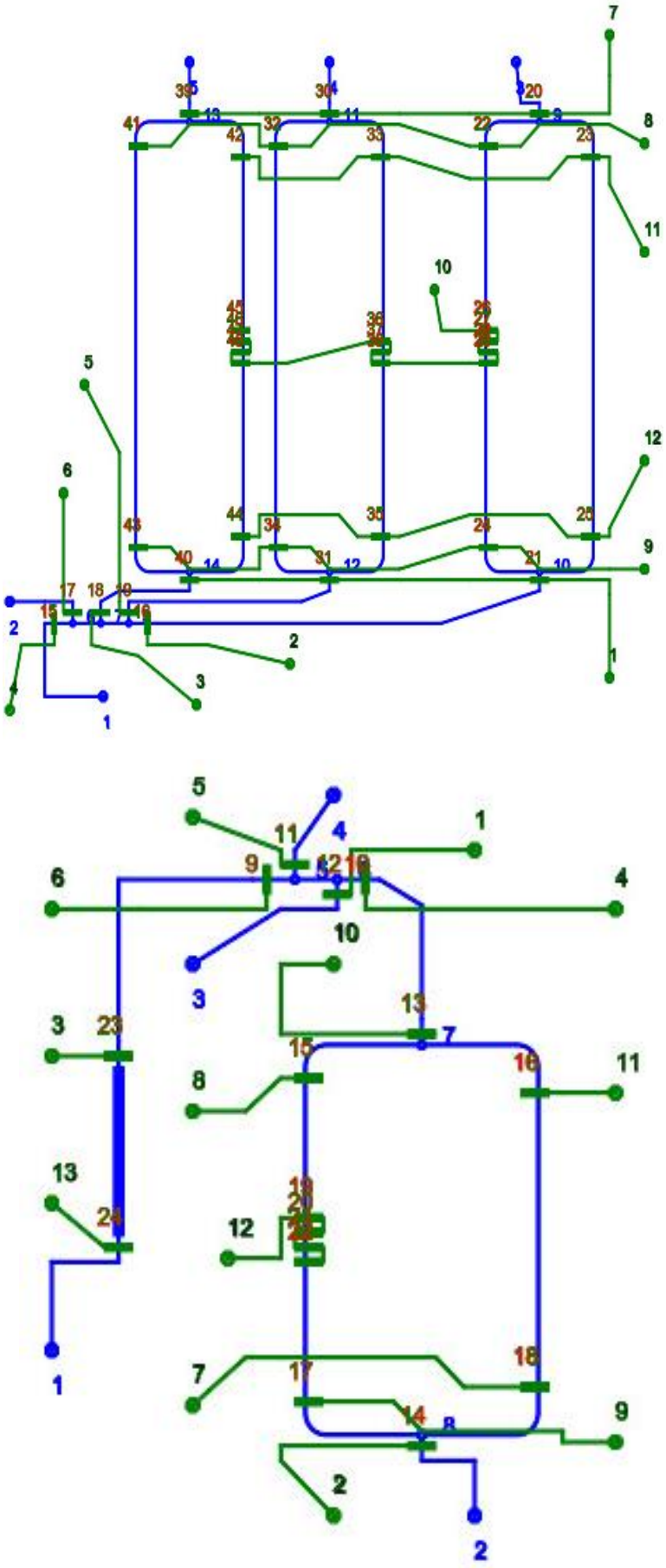


Figure 15: Numbering of all "landmarks". Visualization of Testcase 1(bottom); Visualization of Testcase 2(top)

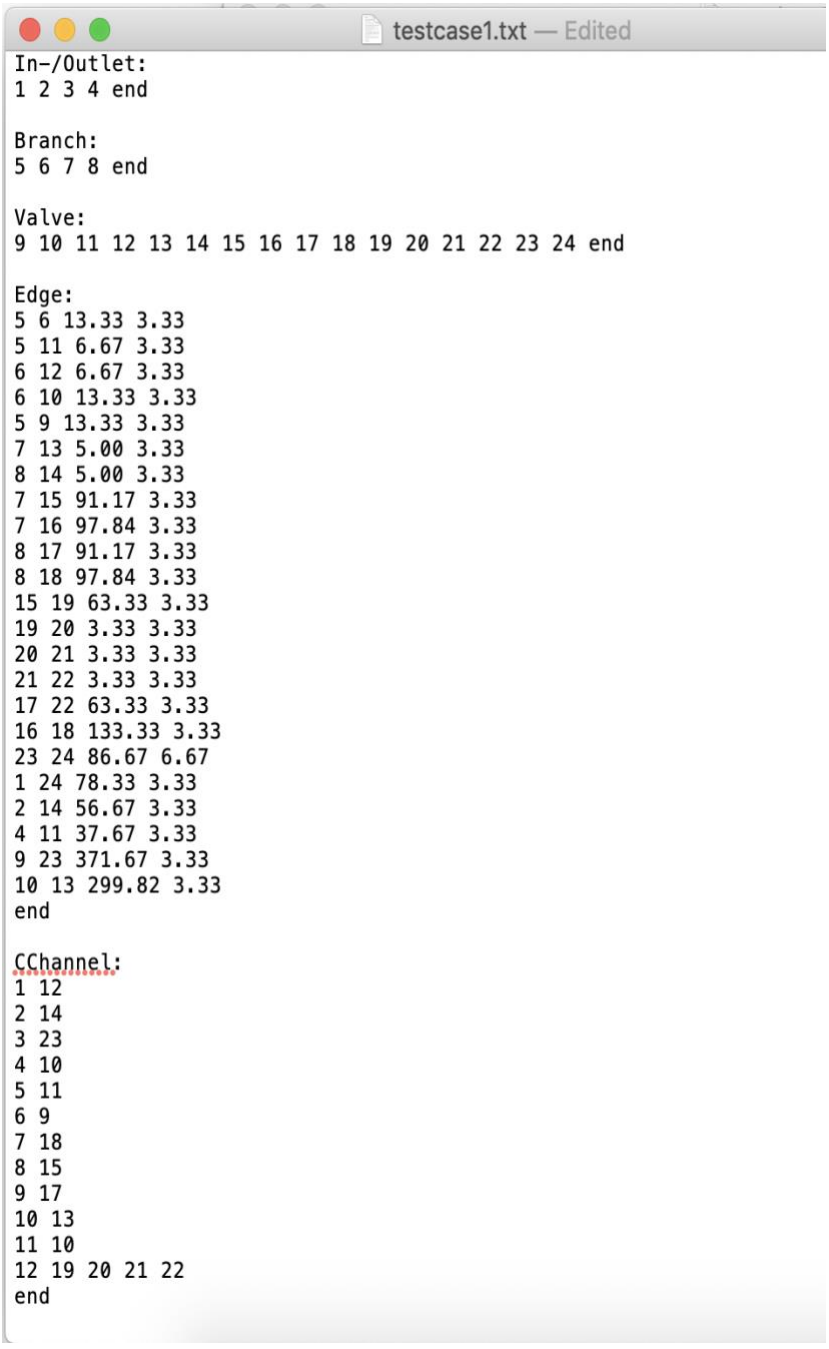


Figure 16: Output text file of Testcase 1

testcase2.txt — Edited

```
In-/Outlet:
1 2 3 4 5 end

Branch:
6 7 8 9 10 11 12 13 14 end

Valve:
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 end

Edge:
6 7 13.33 3.33
7 8 13.33 3.33
6 17 6.67 3.33
7 18 6.67 3.33
8 19 6.67 3.33
8 16 13.33 3.33
6 15 13.33 3.33
9 20 5.00 3.33
10 21 5.00 3.33
9 22 141.17 3.33
9 23 147.84 3.33
10 24 141.17 3.33
10 25 147.84 3.33
22 26 113.33 3.33
26 27 3.33 3.33
27 28 3.33 3.33
28 29 3.33 3.33
24 29 113.33 3.33
23 25 233.33 3.33
11 30 5.00 3.33
12 31 5.00 3.33
11 32 141.17 3.33
11 33 147.84 3.33
12 34 141.17 3.33
12 35 147.84 3.33
33 36 113.33 3.33
36 37 3.33 3.33
37 38 3.33 3.33
38 35 106.67 3.33
32 34 246.67 3.33
13 39 5.00 3.33
14 40 5.00 3.33
13 41 141.17 3.33
13 42 147.84 3.33
14 43 141.17 3.33
14 44 147.84 3.33
42 45 106.67 3.33
45 46 3.33 3.33
46 47 3.33 3.33
47 48 3.33 3.33
48 44 106.67 3.33
41 43 246.67 3.33
1 15 93.33 3.33
2 17 51.67 3.33
3 20 45.22 3.33
4 30 31.67 3.33
5 39 31.67 3.33
16 21 1030.00 3.33
18 40 87.87 3.33
19 31 631.67 3.33
end

CChannel:
1 21 31 40
2 16
3 18
4 15
5 19
6 17
7 20 30 39
8 22 32 41
9 24 34 43
10 26 27 28 29 36 37 38 45 46 47 48
11 23 33 42
12 25 35 44
end
```

Figure 17: Output text file of Testcase 2

5. Conclusion

In this work I propose a program, which automatically marks out all the landmarks required for the bio-protocol simulation in Columba designs. It can connect Columba with the new research for finding the better performance of microfluidic biochips.

6. References

[1]. *Magick*++. (n.d.). Retrieved from ImageMagick Magick++ API: https://imagemagick.org/Magick++/

[2]. Marc A. Unger, H.-P. C. (2000). Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography. *Science, 288*, pp. 113-114.

[3]. Schmidt, A., & Fitzgerald, M. (2017). *Portable, Powerless Automation of Valve Actuation for Microfluidic Large Scale Integration Technology.*

[4]. Setchell, M. (2016). *Stack Overflow*. Retrieved from problems-using-imagemagick-in-xcode-c-application: https://stackoverflow.com/questions/41254653/problems-using-imagemagick-in-xcode-c-application

[5]. Todd Thorsen, S. J. (2002). Microfluidic Large-Scale Integration. *Science, 298*, pp. 580-584.

[6]. Tsun-Ming Tseng, M. L.-Y. (n.d.). Columba 2.0: A Co-Layout Synthesis Tool for Continuous-Flow Microfluidic Biochips. *37*(8).