

EC552/BE552 - Computational Synthetic Biology For  
Engineers

Homework 2  
Spring 2019

March 19, 2020

**BioElectronic Design**

**Due: 04/17/20**

---

---

## Overview

Goal: Form an understanding of the complexity of microfluidic design problems and how computational design tools can be used to build them. In order to help build your skillsets towards lab automation, you will be required to design a device for COVID-19.

The homework is divided into multiple discrete activities, each of the activities will help you understand how to approach the problem step-by-step and work towards the final solution.

Activities:

1. **(75 Points)** Find a published protocol corresponding to COVID-19 research. Your report must include the following information about the protocol:
  - Protocol Motivation (REPORT)
  - Protocol Steps (SEPARATE, REPORT)
  - Protocol Materials (REPORT)
  
2. **(75 Points)** Find a collection (single device when applicable) of published continuous-flow based microfluidic devices that can perform the protocol steps necessary for performing the protocol. Your report must include the following attributes about your devices:
  - Dimensions (REPORT)
  - Parameters (REPORT)
  - Valve control sequences (when applicable) (SEPARATE)
  - Fabrication techniques (REPORT)
  
3. **(75 Points)** Use 3D $\mu$ F to create the device. Deliverables include:
  - Github Pull Requests for new components (SEPARATE)
  - SVG file (SEPARATE)
  - JSON file (SEPARATE)
  - 3D $\mu$ F Render of the Design (REPORT)
  
4. **(75 Points)** Use Neptune to automatically place and route your chosen device.
  - (a) Create a MINT description of the device (REPORT, MINT)
  - (b) Use Neptune to place and route the MINT description. Deliverables include (SEPARATE):
    - SVG design output of Fluigi

- Init\_Parameters.ini file
- (c) LFR Description (Graph) (.lfr) of the Device (SEPARATE, REPORT)
- (d) Fluid Interaction Graph (REPORT)
- (e) Fabricate the Device (If logistics permit)

**Note 1:** Grading will be done on the deliverables so please ensure that all the required deliverables are included to ensure proper grading.

**Note 2:** The final design report submitted must in the format of a Lab on a Chip Communication ([Microsoft Word Template](#), [LaTeX Template](#), [Overleaf LaTeX Templates](#))

---

---

## Activity #1

In this activity you will do a literature search for a laboratory research method and outline the protocol for COVID-19:

- a. Diagnose
- b. Research a vaccine
- c. Generate data for research

An example for a good protocol can be found here: [Sample Protocol](#). Please attach a detailed description of the protocol involved.

The deliverable for **Activity #1** is a report that **clearly** describes the following:

- The protocol or research method being investigated
- The importance of the protocol
- The reagents and corresponding volumes of the protocol (in a table)
- A clear description of the experimental protocol your device was designed to perform

## Activity #2

In this activity you will do a literature search for a set of continuous flow microfluidic device that can either be used to implement the protocols used in the previous activity. The device **must** be based on continuous flow (not electro-wetting/digital-droplet, etc.)). Your device can have valves that are pneumatically actuated. The following journals are a great place to begin your search for an interesting device. This by no means an exhaustive list:

- [Lab on a Chip](#)
- [Microfluidics and Nanofluidics](#)
- [Biomedical Microdevices](#)

In order to make the best decision regarding which device to analyze for this assignment, please read the entire homework assignment before selecting a device. This is to ensure you understand what you will be doing with this device (ex., if you pick a device without valves, it will be impossible to complete many of the activities in this assignment). Points will be awarded based on the quality of the deliverables as well as how interesting and complex the device is (an extremely simple device that does not lend itself well to designing with tools will have a hard time earning top marks in this assignment).

The deliverable for Activity #2 is a report that **clearly** describes the following:

- The device's physical dimensions
- What are some of the device's parameters (channel width, number of inputs, number of valves, etc.)? How does the function of the device change as these parameters change?
- How does this device scale? Can you tile the device to make larger devices? Cascade the device and do intermediate operations?
- A temporal description of the valve control sequence(when applicable).
  - Hint: You should clearly link the physical architecture of the device to the experimental protocol that the device was designed to perform. All intermediate steps performed on the microfluidic chip should be clearly documented here.
- A description of the fabrication technique used to create the device. Popular techniques are soft lithography, CNC milling, wet etching, etc.

---

---

## Activity #3

**3D $\mu$ F** is a user-friendly CAD tool for placing microfluidic components on a grid and exporting the design into a few useful file formats. Your job in Activity #3 is to use 3D $\mu$ F to design the device designed in Activity #2 and export the device in three file formats: JSON (to send to microfluidic control software), SVG (for use with a CNC mill), and STL (for 3D printing).

It is very likely that 3D $\mu$ F [1] will not have all of the components you need to properly design your device. Your job will be to clearly describe features that 3D $\mu$ F cannot draw and create new components for each one that 3D $\mu$ F does not support. You can find the guide to creating new components in Chapter 6 of the **Supplementary Information** of 3D $\mu$ F [1]

Deliverables for Activity #3:

- The report should include an image of the generated render from 3D $\mu$ F
- Include the Github Pull Requests references for new components
- SVG file
- JSON file
- Pull request implementing the new component to 3D $\mu$ F, including primitive definitions for each feature.
  - Significant bonus points will be awarded for implementing the fabrication process in javascript and submitting a pull request to the **3D $\mu$ F Github repository**

## Activity #4

Neptune is a specify, design, and build tool for the development of continuous flow microfluidics. With Neptune, users can specify microfluidic devices through a high level description of liquid flow relations (LFR). The application will automatically generate the physical design of the microfluidic device that can be opened in 3D $\mu$ F for making edits or generating the designs for fabrication. Neptune is currently available at <http://fluigicad.org>.

Neptune comes with the Fluigi tool chain which can take in two kinds of inputs: a) Liquid Flow Relations (LFR) file b) Microfluidic Netlist (MINT) file. In this activity you will create both an LFR file and MINT file that sufficiently describes the microfluidic device.

In order to simplify the design, you can use [Visual Studio Code](#) and install the plugins for [MINT and LFR](#).

### With LFR:

1. Create a description of the of the protocol/device using LFR, Liquid Flow Relations
2. Include comments into the LFR description file.
3. Generate FIG and MINT file from LFR description

### With MINT:

1. Create a description of the of the device using MINT, a microfluidic hardware description language.
2. Include comments into the Mint description file.
3. Place and route the device using Neptune.

**Note:** Since Neptune is actively under development. You might face some problems in using the tool. If you face any problems please make sure you create an Issue on [Github](#), with screenshots and error messages so that they can be addressed quickly.

Deliverables for Activity #3:

1. LFR description of the protocol/device named eg. 'device.lfr'
2. Rendered FIG graph in the report.
3. MINT Description of the microfluidic Device named eg. 'device.uf'.

4. Clearly document the features that could not be implemented using MINT (REPORT) showing images and the alternative components that were used in place of the original feature.
5. Fluigi input parameters used for routing named eg. 'fluigi.ini'.
6. Brief note on the changes incorporated in the PAR/DRC parameters for the device in the report.
7. EPS design file of the generated design.
8. Log generated by Neptune.
9. Fabricated Device

## Dealing with Fluigi/MINT Error

Since Fluigi is a research tool in active development, there might be some constructs or scenarios where the application will not work as expected or might not accurately report the errors in the Mint file.

The binary version of Fluigi that will be distributed will not have some of the actively developed functionality, namely the verilog device generation and the hierarchical design parsing (i.e. the 'IMPORT device' option).

In many cases the Place and Route (PAR) might fail because it does not find a good solution or finds a solution that would violate the Design Rule Check (DRC). In these event's the report should include all the failure messages and the snapshots generated by Fluigi to avail the bonus points and to demonstrate the limitations of the active research tool.

In the scenarios where Neptune crashes or fails, Github issues should created for each of the scenarios on the homework/Fluigi Github repository. This way the questions can be cleared quickly and the binaries can be patched to fix the errors. In addition to that if faced with any other difficulty please contact [sanka@bu.edu](mailto:sanka@bu.edu) requesting assistance/office hours.

## MINT Tutorial

Mint is a microfluidic hardware description language used for describing microfluidic circuits and devices. Just like any other Hardware Description Language like Verilog or VHDL, Mint aims to help microfluidic designers leverage the use of CAD tools by allowing them to describe complex architectures in a human and machine readable format. By using MINT descriptions of the hardware. The designers can easily share and tweak the circuit and component parameters and speed up the entire experimental process.

An example of Mint Syntax can be seen below:

### Mint FAQ

**Q. How can I find the MINT documentation?**

A. MINT's references can be found at [MINT Github Wiki](#).

**Q. How do I create and structure a Mint file and are there any examples?**

A. The basic rules and example mint files can be seen at [Netlist Rules and Examples](#)

**Q. What are the various primitives and modules available in Mint?**

```

DEVICE test01

LAYER FLOW
PORT p1, p2, p3 r=500;
NODE n1;
H LONG CELL TRAP ct1 numChambers=10 chamberWidth=500
chamberLength=500 chamberSpacing=100 channelWidth=500;

CHANNEL c1 from p1 3 to n1 1 w=500;
CHANNEL c2 from p2 1 to n1 3 w=500;
CHANNEL c3 from n1 2 to ct1 1 w=500;
CHANNEL c4 from ct1 2 to p3 4 w=500;
END LAYER

LAYER CONTROL
PORT cp1, cp2 r=500;
VALVE v1 on c1 w=1000 l=500;
VALVE v2 on c2 w=1000 l=500;
CHANNEL c5 from cp1 2 to v1 4 w=500;
CHANNEL c6 from cp2 2 to v2 4 w=500;
END LAYER

```

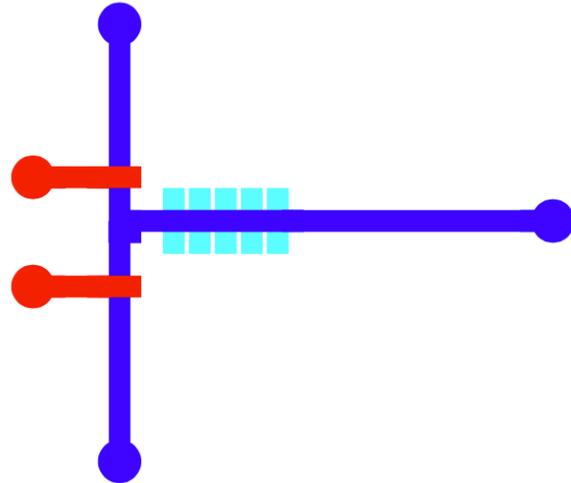


Figure 1: Netlist example. The netlist describes a simple device with three ports and a cell trap on the flow layer and two valves and ports on the control layer. The name of the device is given in the first line. The features and connections on the flow layer are described first, followed by those on the control layer. The design described in the netlist is shown on the right.

A. A complete list of primitives that can be used in in Mint can be found in [the Primitives section of the documentation](#).

**Q. What are the more complex module types available in Mint?**

A. A complete list of predefined modules available in Mint can be found at [Predefined Modules](#)

## Neptune Tutorial

Neptune is a specify design and build tool for Microfluidics. At its core, it contains a microfluidic place and route tool used for laying out the microfluidic components described in a Mint file and properly connecting the components according to the Mint description.

## Neptune FAQ

**Q. How do I get the Neptune binary?**

A. Neptune can be found at [Neptune Github Repo](#)

The Fluigi (Core of Neptune) binary package can be found at [EC500D1 Fluigi Repo](#). Download, place the binaries in your Homework repository and start with the homework.

**Q. What are the dependencies for executing Fluigi?**

A. Fluigi requires JRE 8 for it work. Please make sure that your system has the latest version of JRE installed and working in the command line.

This can be verified by typing the command in your command line :

```
$ java -version
```

**Q. How do I run the Fluigi binary on a Mint file?**

A. This can be seen at [Usage](#)

```
java [-Xmx????m] -jar fluigi mintfilename -i parameterFile  
-o outputformat [-v|h]
```

Java Parameters :

-jar : runs the jar file

[-Xmx????m] : Maximum memory that the application can use.

(???? specifies memory in megabytes)

Fluigi Parameters :

-i / --input : Input parameter file

-o / --output: output format

-v / --hdl : Verilog input (Will be disabled)

-h / --help : Prints out command line usage

**Q. What do I do if my design fails to be generated by Fluigi?**

A. First open an Github issue on the Neptune repository showing the error text, attaching an image of the design, Mint design file and the design parameters that have been changed from the default given. Second refer to the answer given in the question to try and mitigate the issue. **In case of unroutable designs, points will be awarded based on the interactions with the course staff recorded using the Github issues tracker.**

**Q. How do I know if my device is unroutable or if Fluigi has broken?**

A. Below is a table that captures the various reasons the PAR might fail and how to mitigate them.

Reason	Mitigation Strategy
1. The placer does not arrive at a good solution.	1. By rerunning the design multiple times until the PAR gives a solution.
2. The placer arrives at a good solution but the DRC and other physical constraints of the design don't allow for successful routing.	2. By trying to modify the design parameters, DRC constraints in the .ini file and fixing the positions of some of the components.

**Q. Where can I find Fluigi Documentation, examples, etc. ?**

A. The [EC500D1/Fluigi Wiki](#) is mirrored to the internal Fluigi Development Wiki and is the default documentation for running Fluigi.

**Q. Where can I log the bugs for the BONUS POINTS ?**

A. YES ! There are bonus points involved, you can log the issues on [EC500D1/Fluigi Issues](#).

## Other COVID-19 Resources

Discord Community for COVID-19 Discussion : <https://discord.gg/w49dyqX>

Project Inspirations : <https://covidbase.com/>

## References

- [1] Radhakrishna Sanka, Joshua Lippai, Dinithi Samarasekera, Sarah Nemsick, and Douglas Densmore. 3DF - Interactive Design Environment for Continuous Flow Microfluidic Devices. *Scientific Reports*, 9(1), December 2019.