Integration of performance metrics into microfluidic design automation

Version	Accepted manuscript
Citation (published version):	"Sanka, Radhakrishna, Densmore, Douglas. "Integration of Performance Metrics into Microfluidic Design Automation". 11th International Workshop on Bio-Design Automation (IWBDA-19), 2019."

https://hdl.handle.net/2144/37587
Boston University

Integration of Performance Metrics into Microfluidic Design Automation

Radhakrishna Sanka
sanka@bu.edu
Boston University

Douglas Densmore
dougd@bu.edu
Boston University

1 INTRODUCTION

The design of microfluidic devices has the potential to accelerate the discovery and the exploration of engineering paradigms [5]; however, the non-standardized technology stacks coupled with the tedious manual design processes have prevented the complete integration of microfluidic technologies into academic and research settings [10]. While a majority of the research in Microfluidic Design Automation (MDA) tools has been focused on physical design automation and the generation of robust architectures, recent works by Grimmer et al. have explored the possibility of integrating device sizing and architecture generation into the design workflows [2, 3]. However, a significant deficit in these methods is their complete reliance on analytical models which are based on simplifying assumptions that often do not represent the experimentally observed behavior into their formulation. In order to bridge the gap between abstractions provided by MDA tools and the generation of realistic devices, we outline a design flow that enables the integration of performance metrics into the MDA design flows enabled by 3DμF [9], Fluigi [4] and DAFD [7].

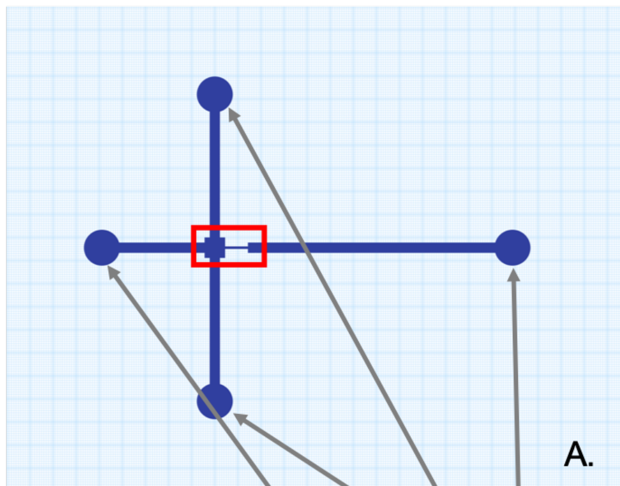
2 PERFORMANCE CHARACTERIZATION

Because of the presence of complex surface interactions and fluid dynamic phenomenon, it is very difficult to characterize the behavior of microfluidic devices. By taking advantage of rapid prototyping techniques [8], DAFD employs machine learning algorithms to characterize and predict microfluidic behavior [6] over a large design space.

In order to use DAFD, the user first needs to generate a dataset that sufficiently characterizes a component by varying the flow conditions and its geometric parameters. After training the performance models within DAFD using the data from characterization experiments, the user would be able to query DAFD for a desired performance alongside the geometric parameters and their respective flow conditions.

3 SPECIFICATION OF PERFORMANCE METRICS

Liquid Flow Relations (LFR) is a hardware description language (HDL) that borrows syntax and concepts from Verilog [1] for describing devices that perform liquid manipulations. LFR allows the designer to abstract the technologies that are used for performing the fluid manipulations and describe the entire device's behavior in terms of how the different fluids that would be injected onto the device would be distributed and controlled.




```
1   module dropletgenerator(waterin, oilin1, oilin2, dropletout);
2
3   finput oilin1, oilin2, waterin;
4   foutput dropletout;
5
6   # MAP 'DROPLET GENERATOR' &
7   # DROPLET_RATE = 30 Hz & DROPLET_SIZE = 85 um
8   assign dropletout = oilin1 & oilin2 & waterin ;
9
10  endmodule
```

B.

Figure 1: Specifying Droplet Generator performance using LFR - A. Shows a representative design of a device that contains a droplet generator. B. Gives the LFR (Liquid Flow Relations) description of the device shown in A. Lines 6-7 how the user can define custom performance constraints while Lines 1 and 8 show how different elements of the device description allow for the synthesis of the design architecture.

Figure 1 shows an example of an LFR specification that is used to describe a device used to generate droplets. Lines 1 and 10 give a top level description where the user explicitly specifies the inputs and outputs. In order to assign the performance specification, the user needs to annotate the *assign* statement (Line 8) to help LFR synthesize a *DROPLET GENERATOR* with the target performance metrics (Lines 6 and 7).

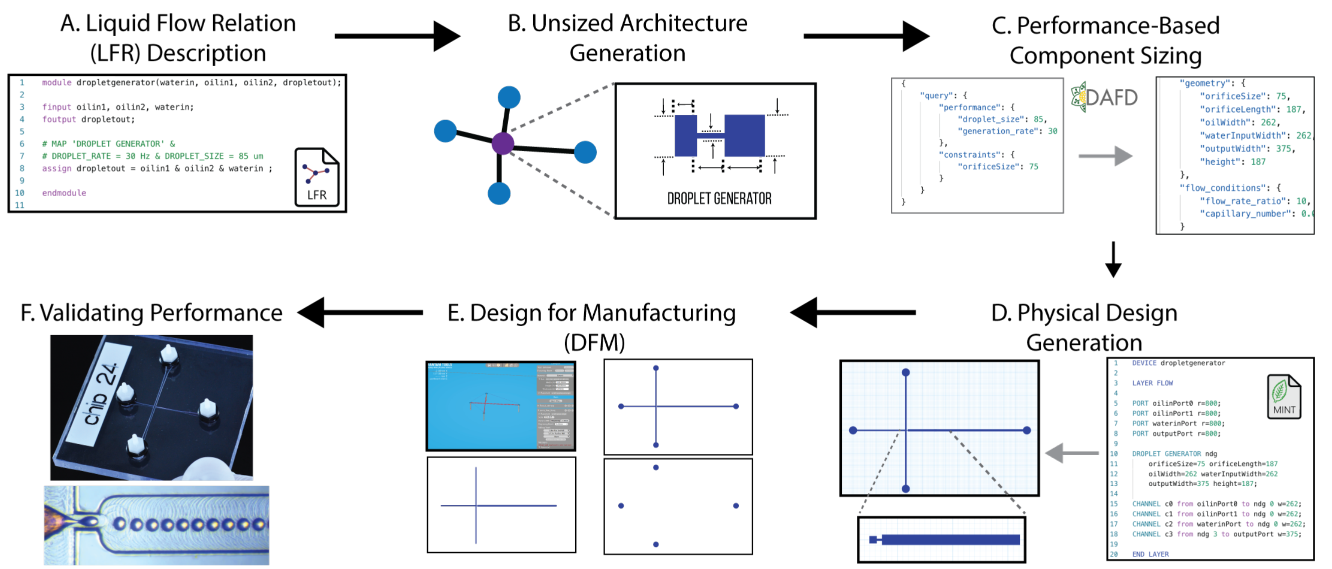


Figure 2: Steps A-F describe how a text based Liquid Flow Relation (LFR) file is converted into a device that matched the performance specification described in the LFR file. The LFR compiler reads the description and creates an unsized architecture (incorrect dimensions). The compiler then queries DAFD with the performance specifications and retrieves component parameters to generate the final design description that is used by Fluigi[4] to generate the layout of the device and the subsequently generate the DFM output that are optimized for different manufacturing processes to be manufactured and validated.

4 ARCHITECTURE SYNTHESIS

In order to generate the microfluidic architecture from LFR specification, the compiler first constructs a *Fluid Interaction Graph* that captures the interactions and behaviors between the various fluid inputs. The compiler then processes the graph to evaluate the logical expressions that are associated with the control inputs and finally maps microfluidic technologies to generate a netlist G that describes the architecture of the device. Where $G \in (V, E)$ is a graph where V is collection of components and E is a collection of connections that link various components.

Once the initial netlist is generated, the LFR compiler sequentially goes through each of the performance specifications associated with netlist and queries DAFD for the geometric parameters that are necessary for individual components to meet their respective performance targets.

5 CONCLUSION AND FUTURE WORK

The integration of performance metrics into the larger MDA workflows (Architecture Synthesis and Physical Design Automation) has the potential to reduce the amount of microfluidic expertise needed to takes ideas seen in literature and apply them in different applications. By expanding the datasets utilized by DAFD to include different microfluidic design primitives, we can extend the capabilities and the efficacy of MDA tools and allow for new synergies between researchers working in microfluidics research and design automation.

REFERENCES

[1] IEEE 1800-2017 - IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language.

[2] Grimmer, A., Haselmayr, W., Springer, A., and Wille, R. A discrete model for Networked Labs-on-Chips: Linking the physical world to design automation. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)* (June 2017), pp. 1–6.

[3] Grimmer, A., Haselmayr, W., and Wille, R. Automated Dimensioning of Networked Labs-on-Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2018), 1–1.

[4] Huang, H. *Fluigi: An end-to-end Software Workflow for Microfluidic Design.* PhD thesis, Boston University, 2015.

[5] Huang, H., and Densmore, D. Integration of microfluidics into the synthetic biology design flow. *Lab on a Chip 14*, 18 (Aug. 2014), 3459–3474.

[6] Lashkaripour, A., Rodriguez, C., Ortiz, L., and Densmore, D. Performance tuning of microfluidic flow-focusing droplet generators. *Lab on a Chip 19*, 6 (Mar. 2019), 1041–1053.

[7] Lashkaripour, A., Sanka, R., Lippai, J., and Densmore, D. Design automation based on fluid dynamics, 2017.

[8] Lashkaripour, A., Silva, R., and Densmore, D. Desktop micromilled microfluidics. *Microfluidics and Nanofluidics 22*, 3 (Mar. 2018), 31.

[9] Sanka, R., Lippai, J., Samarasekera, D., Nemsick, S., and Densmore, D. 3dμf - interactive design environment for continuous flow microfluidic devices. *Scientific Reports (In submission)* (2019).

[10] Whitesides, G. M. The origins and the future of microfluidics. *Nature 442*, 7101 (July 2006), 368–373.