Flow-Based Biochips: Fault-Tolerant Design and Error Recovery

Paul Pop, *Member, IEEE*

I. Background and Motivation

The focus of this paper is on continuous-flow biochips, where the basic building block is a microvalve. By combining these microvalves, more complex units such as mixers, switches, multiplexers can be built, hence the name of the technology, "microfluidic Very Large Scale Integration" (mVLSI) [1]. Biochips are currently being designed manually using tools such as AutoCAD. Physical defects can be introduced during the fabrication process, which reduces the yield, and may lead to the failure of the biochemical application. Failure is costly because of the need to redo lengthy experiments, using expensive reagents and often hard-to-obtain samples, and can be safety critical (endangering human life), e.g., for important diagnostic procedures (screening for cancer). Researchers have started to propose fault models and test techniques for mVLSI biochips [2].

To increase the yield, and to potentially also prevent the failure during the operation of the biochip, we advocate the use of fault-tolerant biochip design. The vision is to provide application fault-tolerance at run-time (online), detecting the faults as they appear, and reconfiguring the application. However, in this paper our assumption is that the faults are detected during testing, and that the operation of the biochip is reconfigured offline (at design time) to avoid the faults. We are interested to introduce redundancy such that the applications can still successfully run on a defective biochip. *Redundancy* is the addition of extra resources, normally not needed for correct operation, to be used for fault-tolerance.

II. Fault-Tolerant Design

We propose a fault-tolerant design strategy, which is part of an overall mVLSI physical design flow described in [3]. Our algorithm takes as input (i) a *netlist* of components, i.e., the components in the architecture and their interconnections, (ii) an application model consisting of a sequencing graph, where each node is an operation and edges capture fluid dependencies, (iii) a fault model, and (iv) a set of constraints imposed by the designer, and produces as output a fault-tolerant netlist. We are interested in that fault-tolerant netlist, which fulfills the constraints imposed by the designer (e.g., in terms of maximum biochip area to be used for fault-tolerance) and corresponds to an architecture that is able to successfully run the biochemical application even in case of the occurrence of faults in the given fault model.

Fig. 1a presents an example input netlist. If the designer has used the biochip extensively and has noticed a repeating fault pattern, such a fault pattern can be provided as input. For example, in Fig. 1a the designer has specified a stuck channel (depicted with a thick red line) and a malfunctioning valve in the pump component of $Mixer_1$. Fig. 1b shows a possible architecture that would tolerate the faults from Fig. 1a. We have introduced a redundant channel (the thick green line), which can be used as an alternative if the channel fails, and we have used a fault- tolerant mixer, i.e., $FT\text{-}Mixer_1$. Such a fault tolerant mixer, uses a fourth valve in the pump component of the mixer, which is normally composed on three valves. Thus, if one of the valves fails, the fault- tolerant mixer still has three functioning valves to perform the needed pumping action.

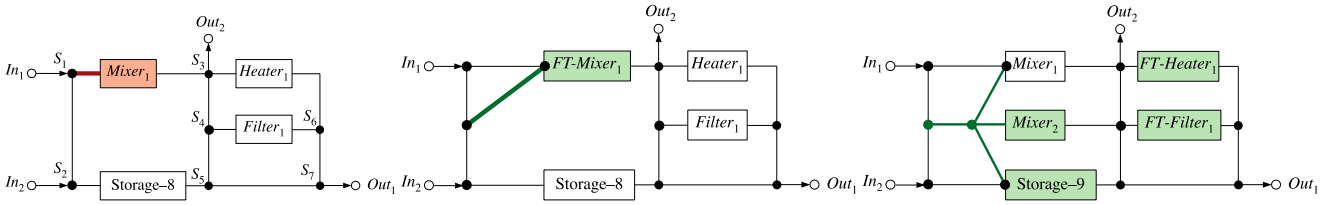
However, often, we do not know the exact fault pattern that has to be tolerated. Instead, the designer would specify a more general fault model. The difficulty in determining a fault-tolerant architecture in this case, is that we do not know a-priori where the fault will actually occur. We know the faults only after we have tested the biochip, and not during the design phase, which is discussed here. A possible such fault-tolerant architecture is presented in Fig. 1c.

References

[1] I. E. Araci and S. R. Quake. 2012. "Microfluidic Very Large Scale Integration (MVLSI) With Integrated Micromechanical Valves". Lab on a Chip 12 (16): 2803-2806.

[2] K. Hu, F. Yu, T. Y. Ho and K. Chakrabarty. 2014. "Testing of Flow-Based Microfluidic Biochips: Fault Modeling, Test Generation, and Experimental Demonstration". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 3(10):1463-1475

[3] W. H. Minhass, P. Pop, J. Madsen and F. S. Blaga. 2012. "Architectural Synthesis of Flow-Based Microfluidic Large-Scale Integration Biochips". Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems: 181-190



(a) Example architecture (netlist) (b) Fault-tolerance to the fault pattern in (a) (c) Fault-tolerant architecture

Figure 1. Fault-tolerant architecture examples

Assoc. Prof. Paul Pop is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark (phone: +45 50 18 62 96; fax: +45 45 88 26 73; e-mail: paupo@dtu.dk).