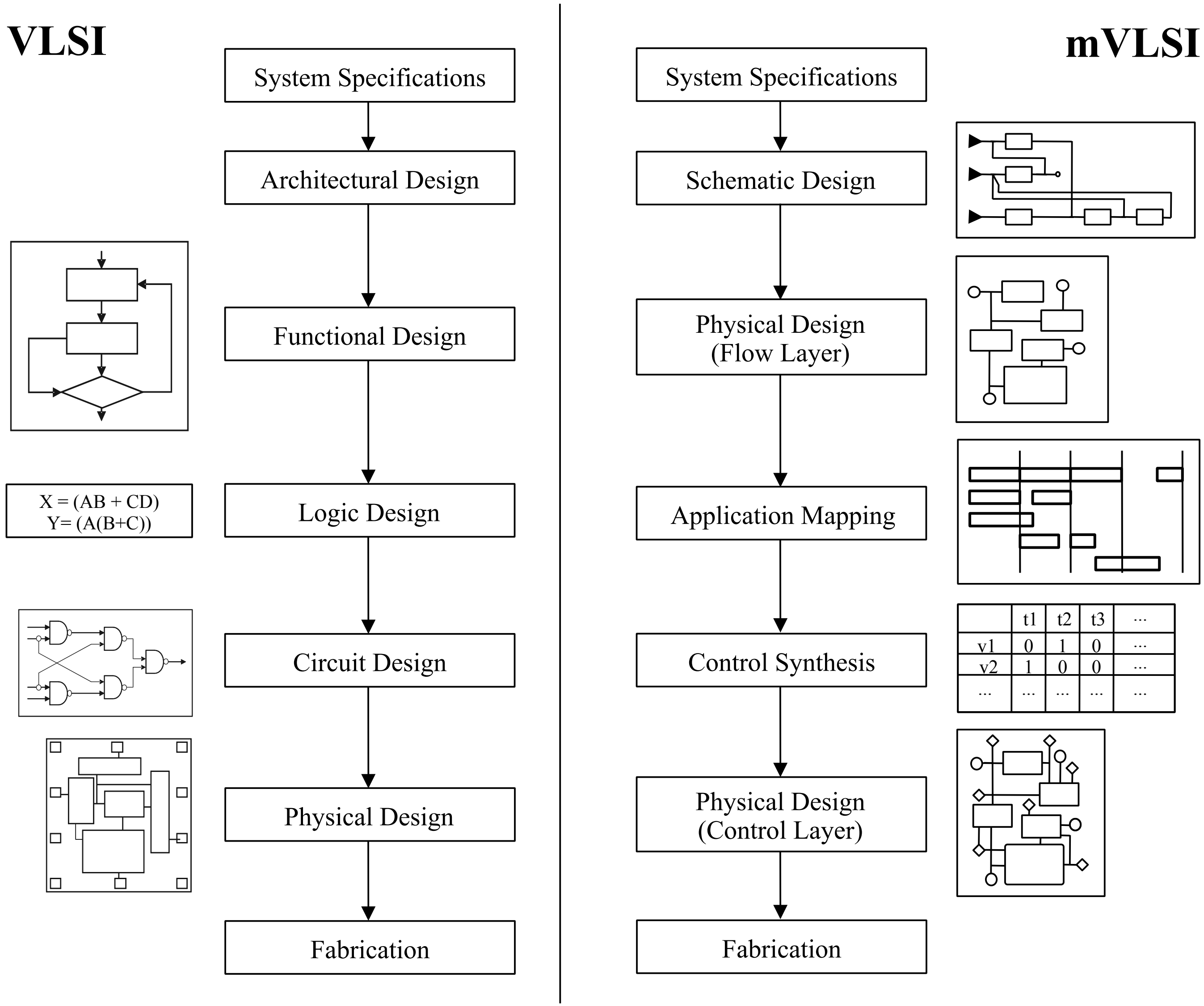
Flow-Based Biochips: Fault-Tolerant Design and Error Recovery

Paul Pop

Technical University of Denmark

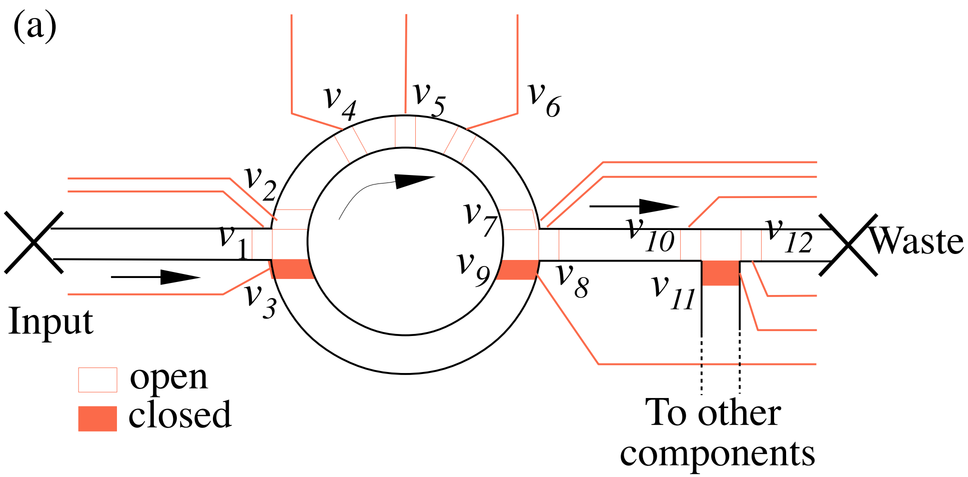
Physical design: VLSI vs mVLSI

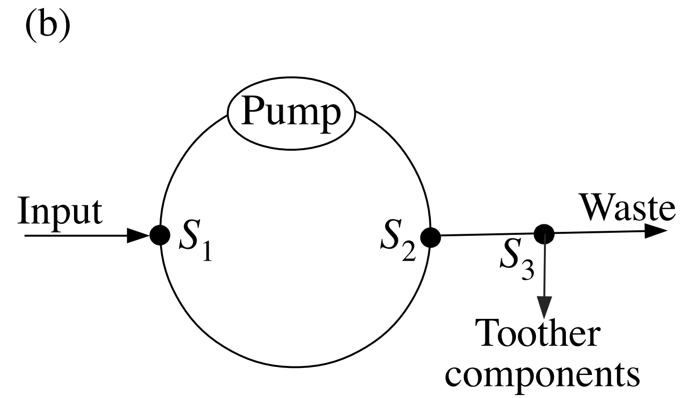


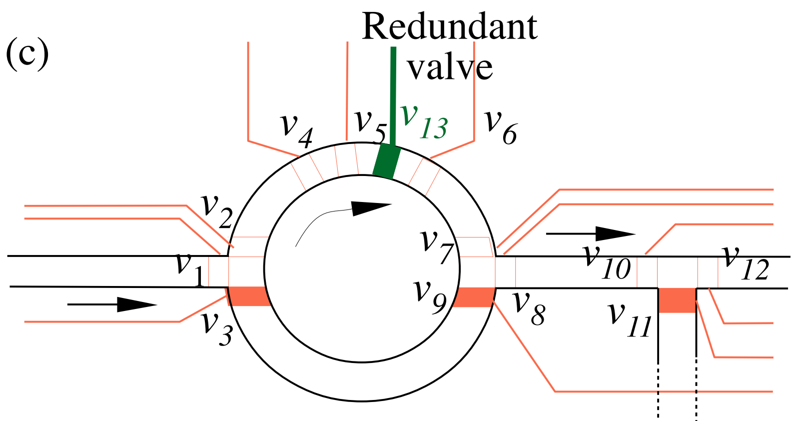
Problem formulation

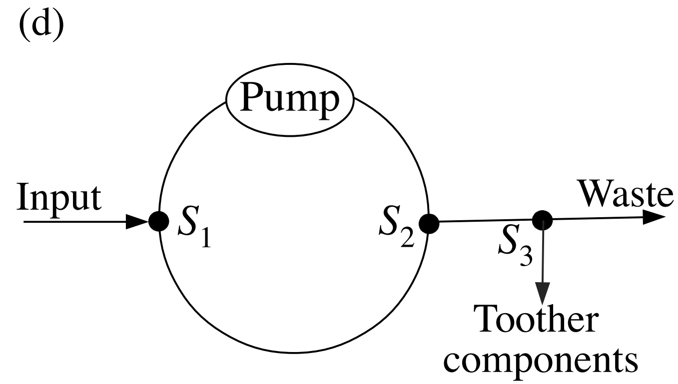
- **Given**
  - A biochemical application (and a fault model)
  - Characterized component model library (including fault-tolerant components)
- **Synthesize**
  - A biochip architecture
  - Deciding on:
    - Component **allocation**
    - Schematic design and (and a fault-tolerant) ***netlist*** generation
    - Physical synthesis
      - **Placement** of components
      - **Routing** of microfluidic channels
  - **Such that**
    - the application completion time is minimized
    - Satisfying the fault-tolerance, dependency and resource constraints

Fault-tolerant components



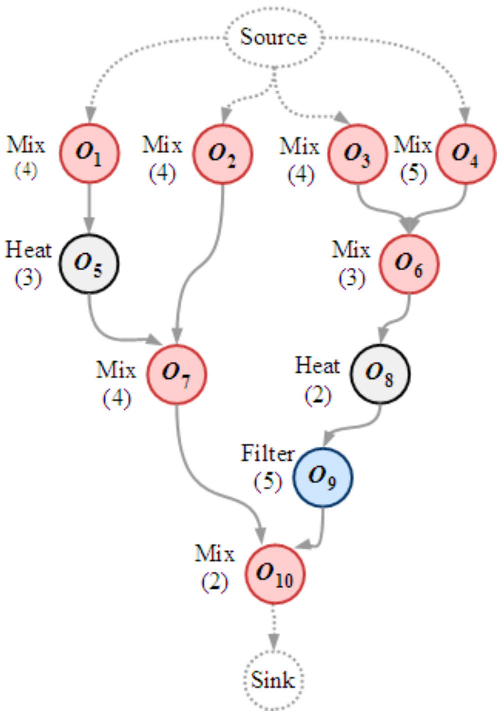




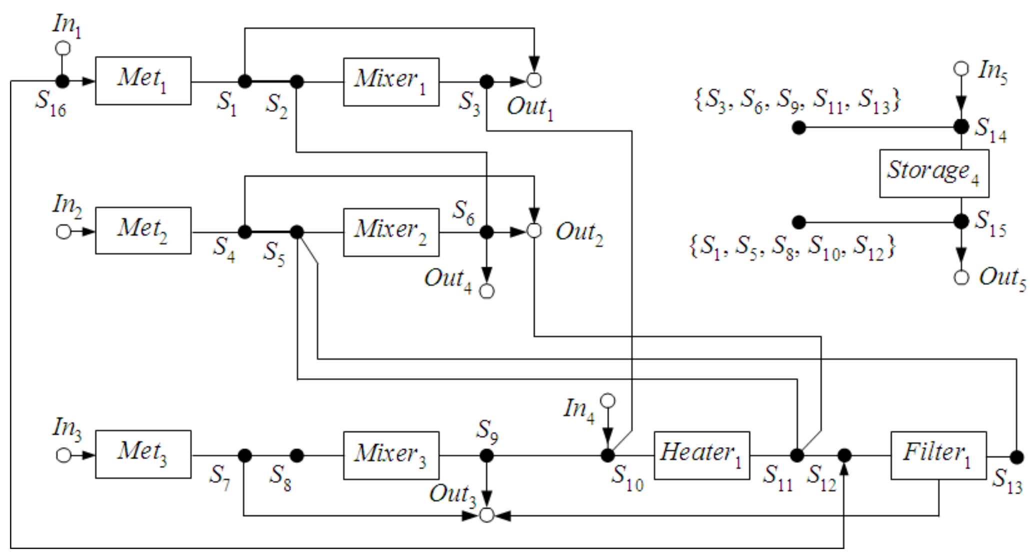


Allocation and schematic design

- How many components, and how to interconnect them?

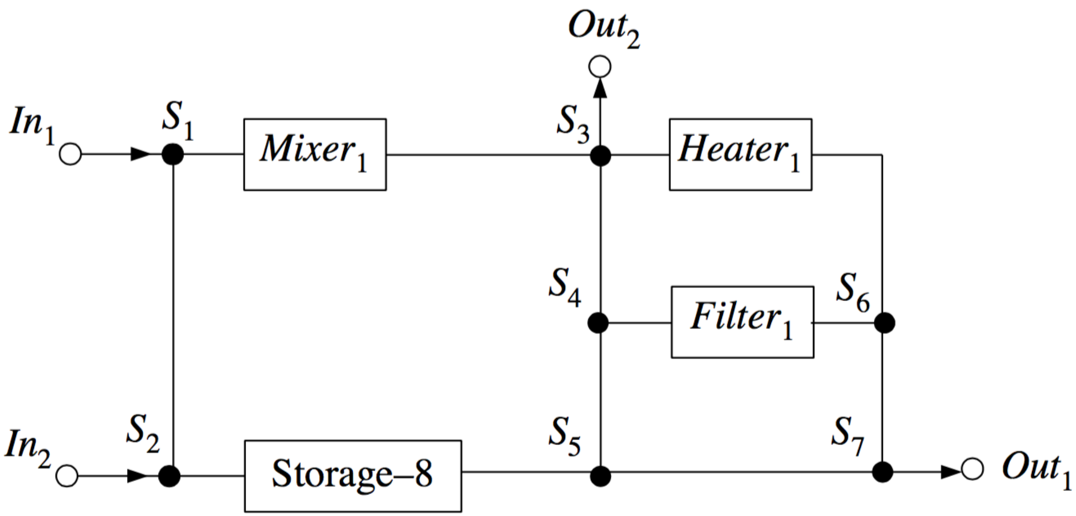


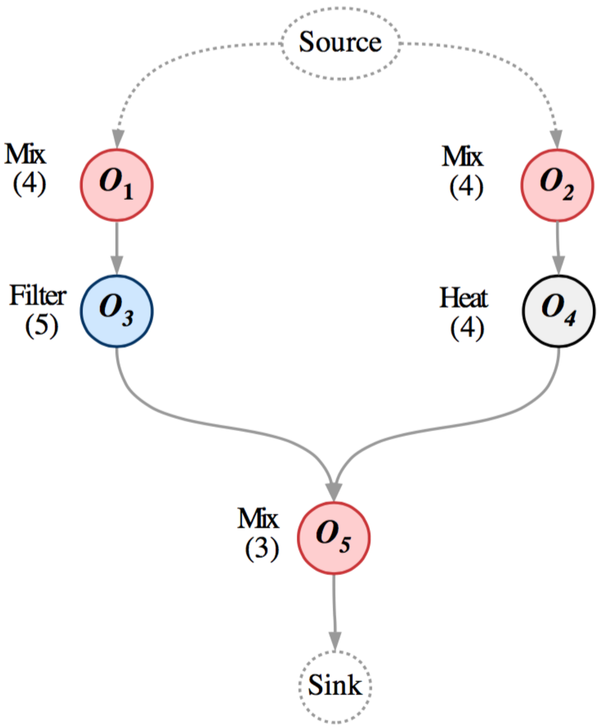
Mixer	3
Heater	2
Filter	1



- Input/ output ports
- Storage units
- Fluidic constraints

Design for fault-tolerance: motivation example





Architecture without fault-tolerance

Application

Fault model

- The designer gives the fault-model as an input: a set of possible faults; any combination may happen

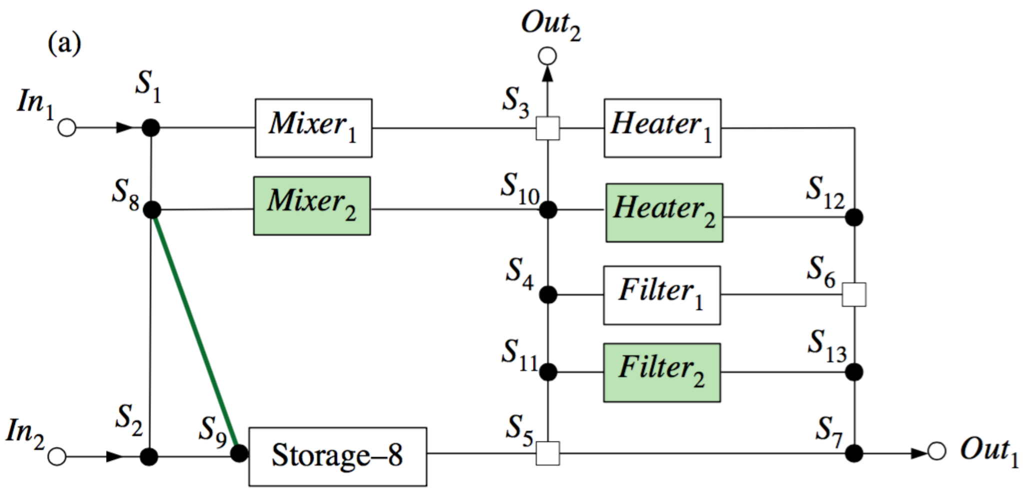
Table: The set of valve faults $\mathcal{VF}$

Name	Vertex ($N \in \mathcal{N}$)	Valve affected (w)	Type (t)
VF_1	$Mixer_1$	v_5	Open
VF_2	S_6	v_3	Open
VF_3	S_5	v_2	Open
VF_4	S_3	v_3	Open

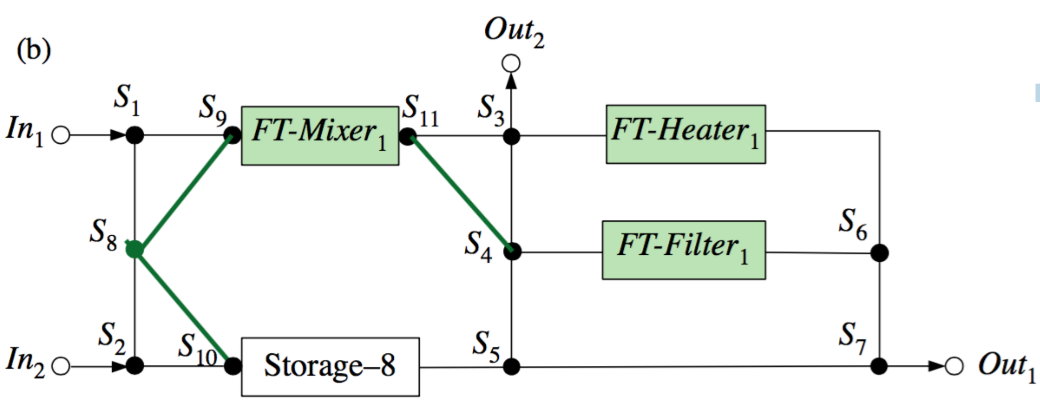
Table: The set of channel faults $\mathcal{CF}$

Name	Component ($M \in \mathcal{N}, \notin \mathcal{S}$) / Connection $D_{i,j} \in \mathcal{D}$	Type (t)
CF_1	$Heater_1$	Block
CF_2	$Filter_1$	Block
CF_3	$S_2 \rightarrow$ Storage-8	Block
CF_4	$S_1 \rightarrow Mixer_1$	Block

Straightforward vs. optimized redundancy



- Straightforward solution: redundancy not optimized; architecture cost: 129



- Optimized solution the introduction of redundancy is optimized; architecture cost: 96

Strategy and evaluation

- Metaheuristic optimization:
  Greedily Randomized Adaptive Search Procedure (GRASP)
  - Searches the solutions space to minimize the objective function
  - Fault scenario generation:
    subset of all the possible scenarios, because their number is huge
    - Each iteration visits a possible solution
      - Applies a fault scenario: injects the faults in the scenario
      - Determines connectivity: can I still move fluids around?
      - Finish time of the application: will the application finish correctly?

- Evaluation: can we obtain a good yield?

$\mathcal{A}_{initial}$			$\lvert\mathcal{FS}\rvert$	$\mathcal{A}_{result}$				$\lvert FT\rvert$	$\lvert\neg FT\rvert$	$FT\%$
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{D}\rvert$	Cost		$\lvert\mathcal{N}\rvert$	FT-$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{D}\rvert$	Cost			
15	17	5760084	25	16	2	20	98	105	16	86.78
15	17	10540084	50	15	3	19	99	117	4	96.69
15	17	18580084	85	16	2	21	101	121	0	100
15	17	27610084	121	15	3	19	99	121	0	100