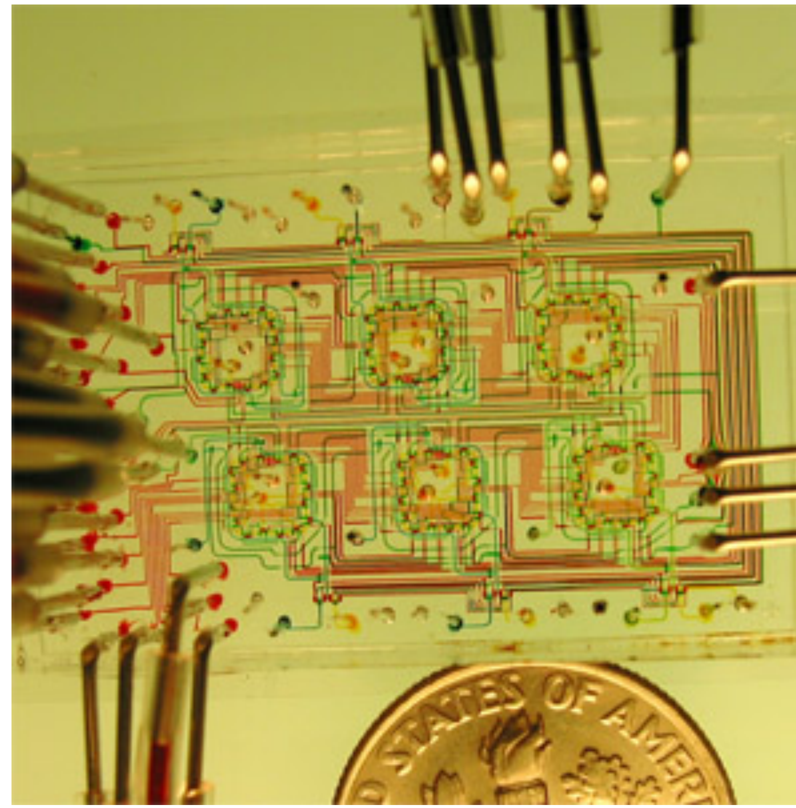
Fault-Tolerant Architecture Design for Flow-Based Biochips

Morten Chabert Eskesen

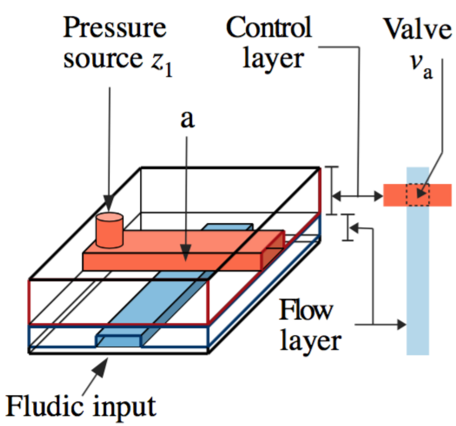
DTU Compute
Department of Applied Mathematics and Computer Science

Flow-Based Biochips

- Can run biochemical applications
  - Microreaction Technology
  - Cell Biology
  - Diagnosis Testing



Basic building block



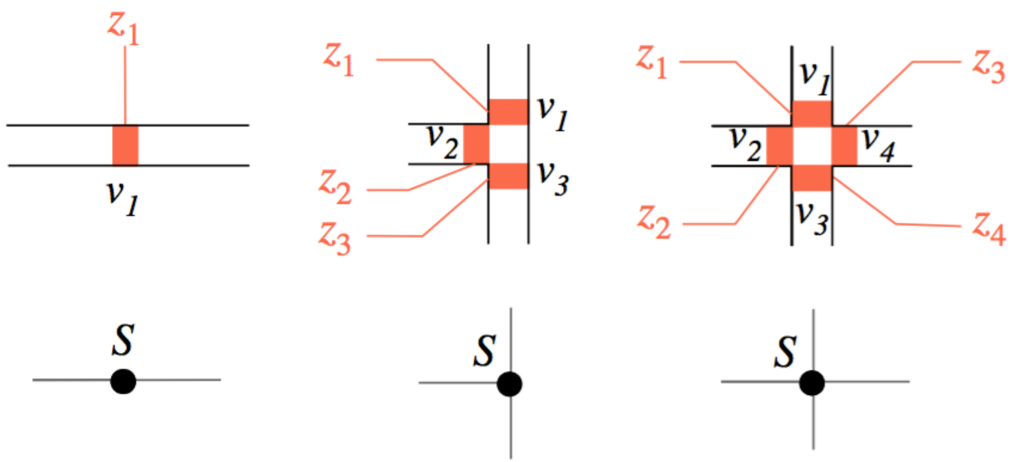
Outline

- Components
- Faults, Causes and Fault Modeling
- Component Library Extension
  - Fault-tolerant components
- Problem Formulation
- Motivational Example
- Architecture Evaluation
- Architecture Synthesis
  - Design Transformations
- Proposed Solutions
- Experimental Evaluation
- Conclusions

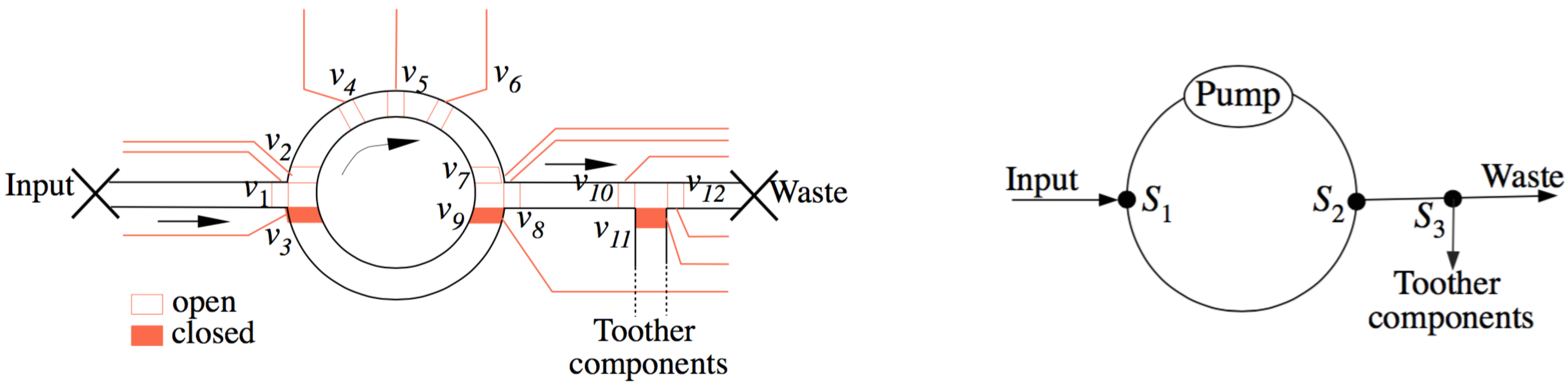
Components

Valves combine to form more complex units

Microfluidic Switch

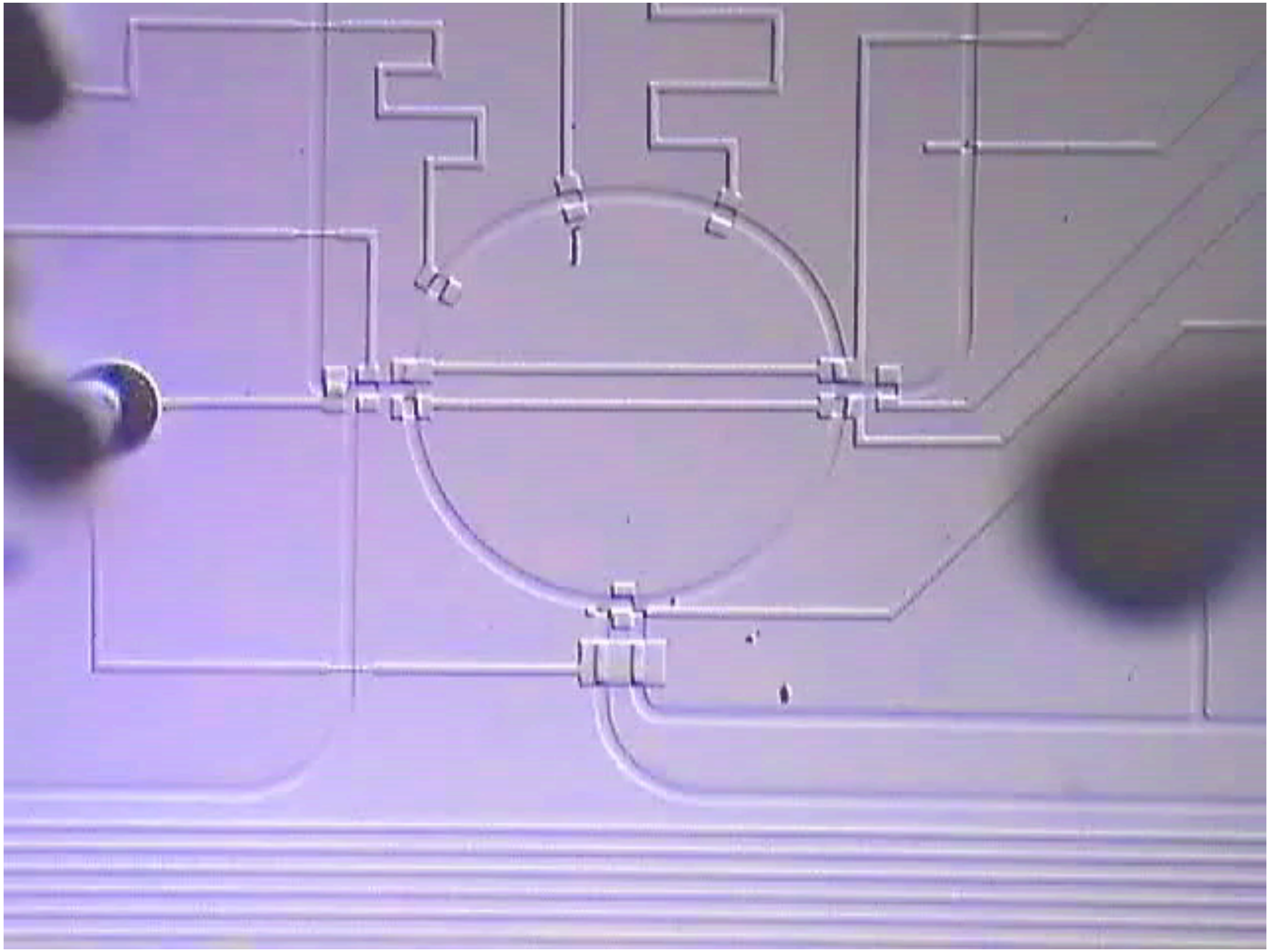


Microfluidic Mixer



Video of Mixer

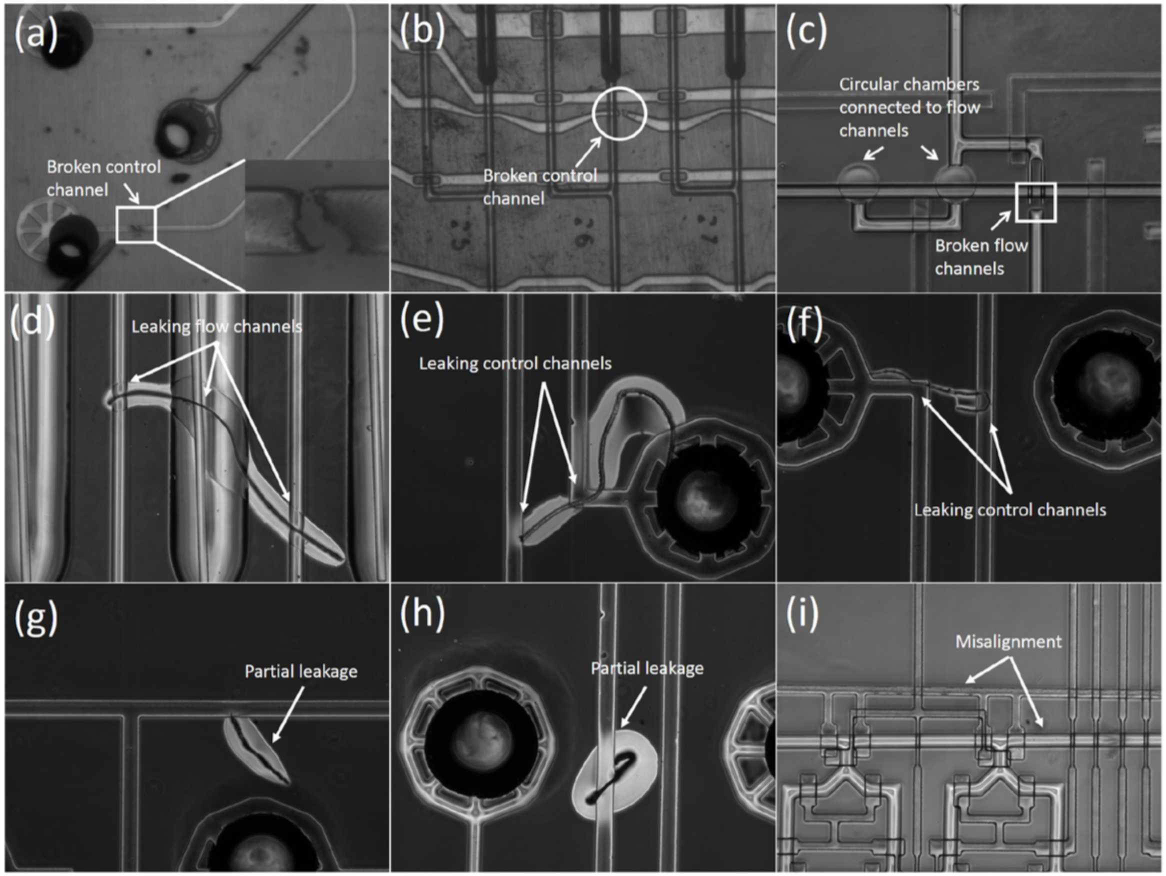
Microfluidic Mixer



http://groups.csail.mit.edu/cag/biostream/

Faults, Causes and Fault Modeling

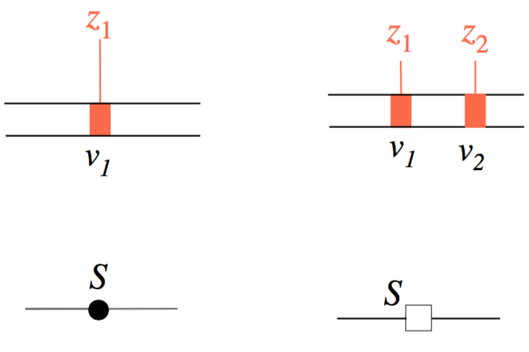
- Block (a)-(c)
- Leak (d)-(f)
  - Partial leak (g)-(h)
- Misalignment (i)
- Faulty pumps
- Degradation of valves
- Dimensional errors



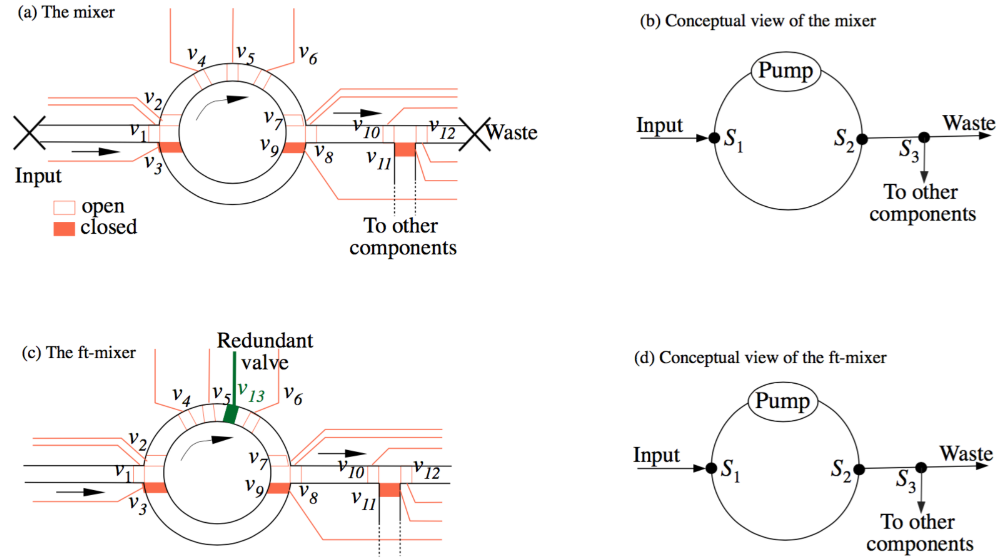
	Flow Layer	Control Layer
Block	Valve is stuck closed	Valve is stuck open
Leak	Fluid flow in one channel contaminates adjacent channels	Control channels of two independent valves are unintentionally connected. Pressure on either valve closes both valves

Component Library - Extension

Switch



Mixer

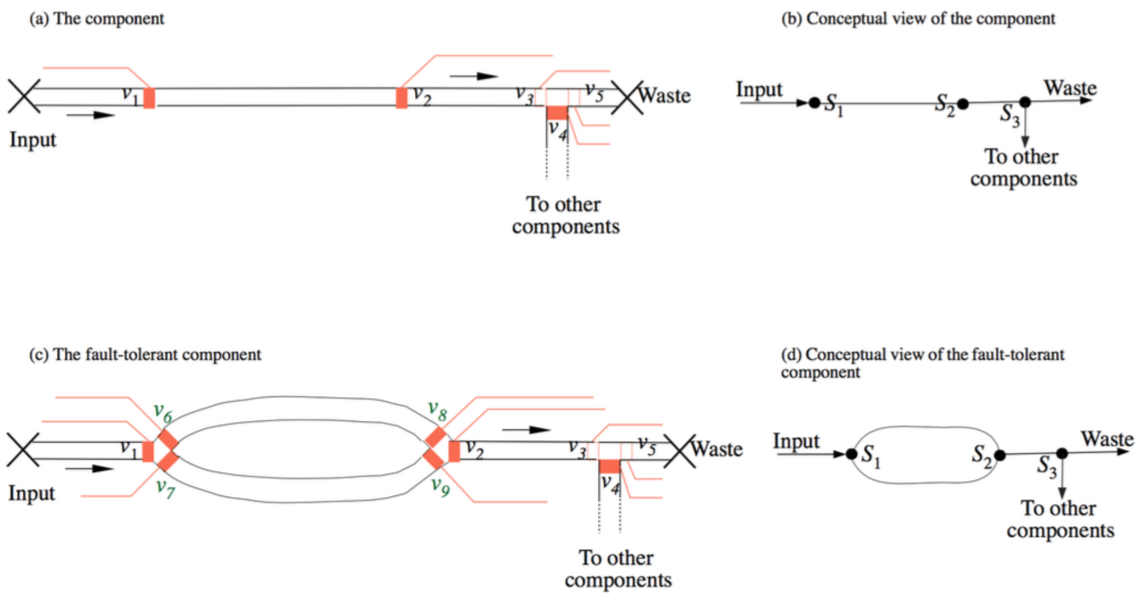


Heater

Detector

Filter

Separator

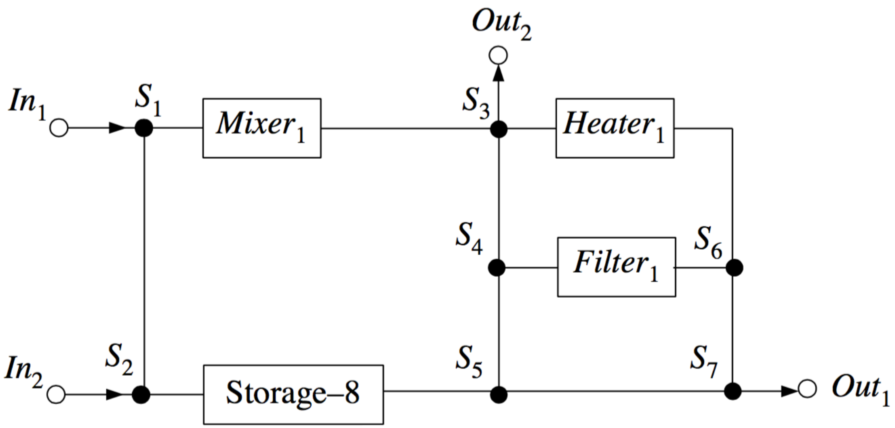


Problem Formulation

- Given
  - A netlist (components and their interconnections)
  - A component library *L*
  - An application graph *G* with a deadline
  - A fault model *Z*
- Determine
  - A fault-tolerant netlist
- Such that
  - The biochip is fault-tolerant and
  - It minimises the cost of the architecture

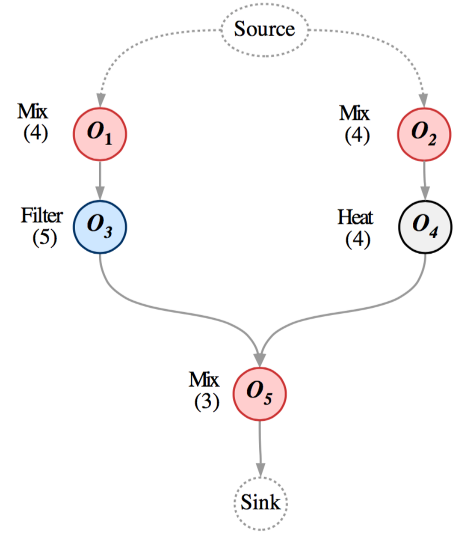
Motivational Example - Setup

Initial architecture



$\mathcal{A}^{initial}$

Application Graph



$\mathcal{G}(\mathcal{O},\mathcal{E})$

$$\mathcal{Z} = (\mathcal{VF},\mathcal{CF},v,c)$$

$$\mathcal{Z} = (\mathcal{VF},\mathcal{CF},2,2)$$

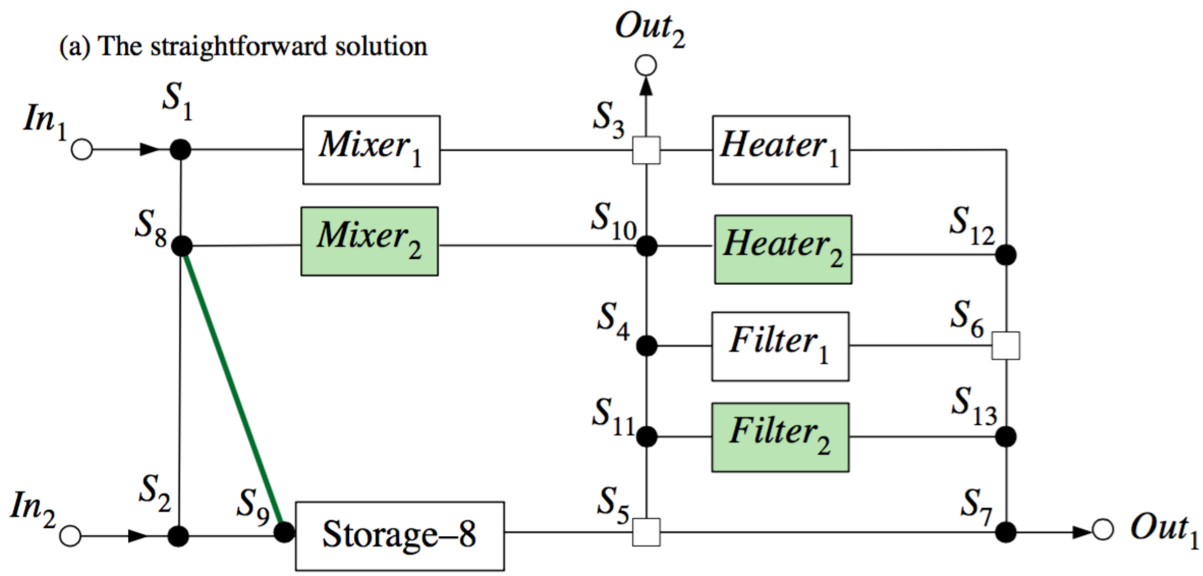
Name	**Component** $(M \in \mathcal{N}, \notin \mathcal{S})$ / **Connection** $D_{i,j} \in \mathcal{D}$	**Type** (t)
CF_1	$Heater_1$	Block
CF_2	$Filter_1$	Block
CF_3	$S_2 \rightarrow$ Storage-8	Block
CF_4	$S_1 \rightarrow Mixer_1$	Block

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

Motivational Example - Solution

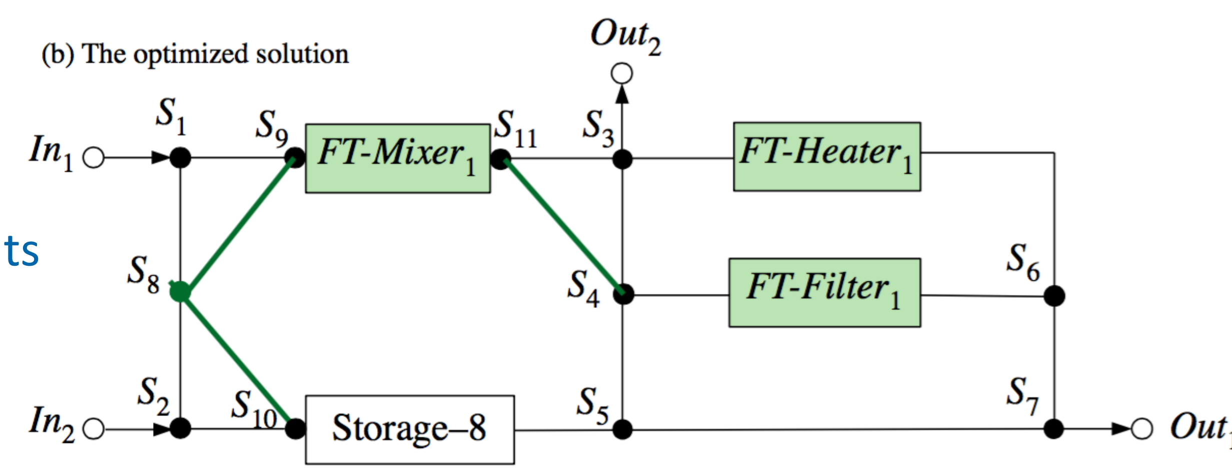
- Straightforward solution
  - Redundant components
  - Cost: 129

(a) The straightforward solution



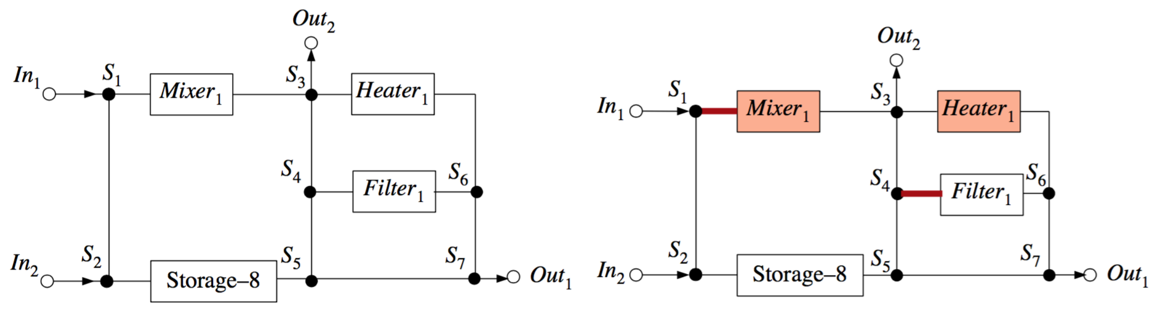
- Optimised solution
  - Fault-tolerant components
  - Cost: 96

(b) The optimized solution



Architecture Evaluation

- Fault scenario generation
  - Each iteration
    - Applies a fault scenario
    - Determines connectivity
    - Finish time, δ, of the application



(a) $A^{original}$ (b) A affected by fault scenario

Channel $S_1 \rightarrow Mixer_1$ is blocked, Channel of $Heater_1$ suffers from a block defect.

A valve in the pump of $Mixer_1$ and the valves controlling the channel towards S_3 and the channel towards S_5 of S_4 are stuck open

$$Objective(\mathcal{A}) = \left(\sum_{f \in \mathcal{FS}}^{\mathcal{FS}} \neg ft \right) \times W_{ft} + \left(\sum_{f \in \mathcal{FS}}^{\mathcal{FS}} max(0, \delta - d_{\mathcal{G}}) \right) \times W_s + Cost_{\mathcal{A}}$$

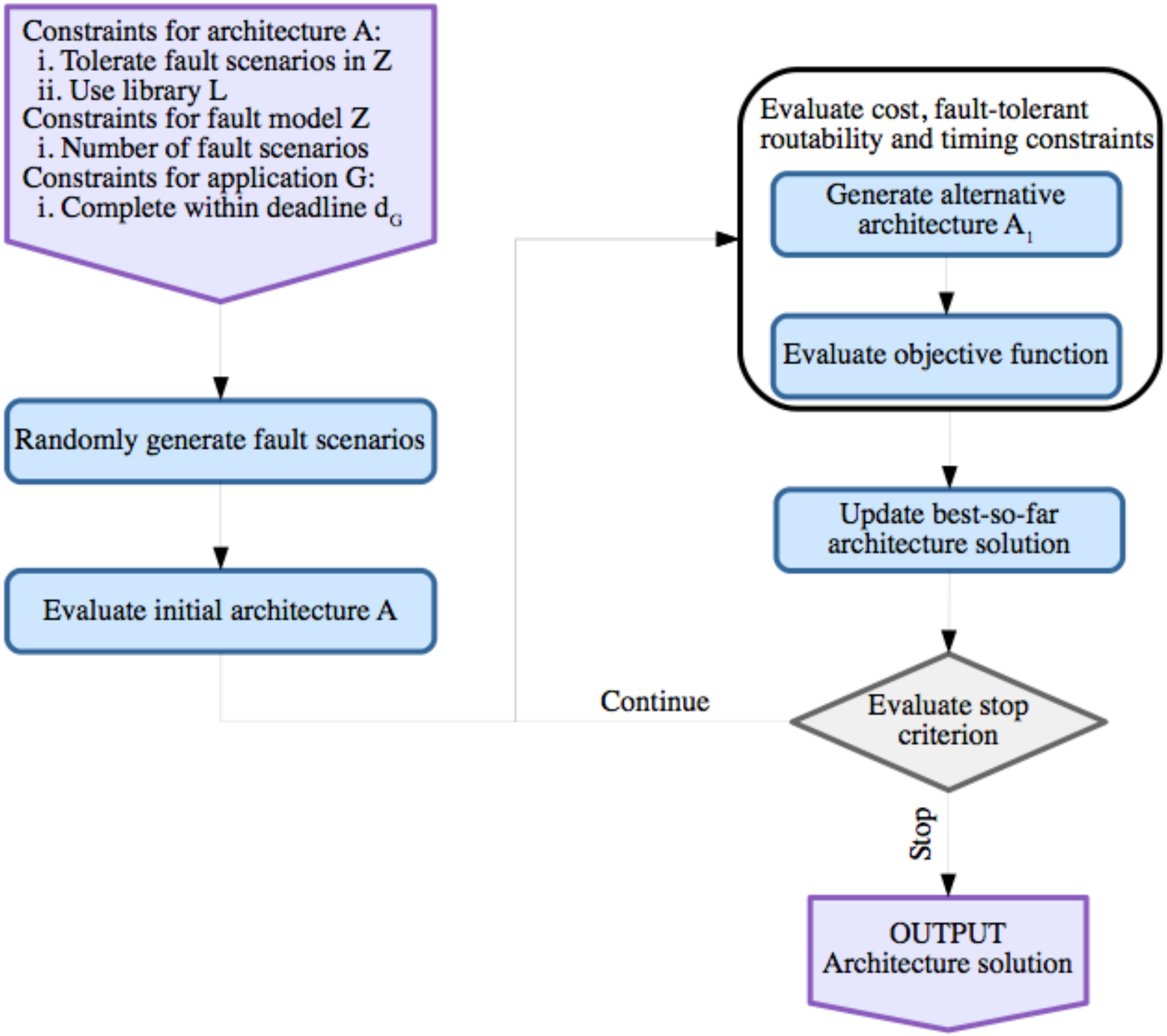
- Connectivity
  - 1 if connected, 0 otherwise
  - Determined by Breadth First Search
  - Considers blocked route and valve faults on switches

- Scheduling
  - Maximum of 0 or δ - application deadline
  - Determined by List Scheduling
  - Routes determined by Breadth First Search

- Physical constraints
  - Sum of
    - Total number of valves
    - Total number of connections

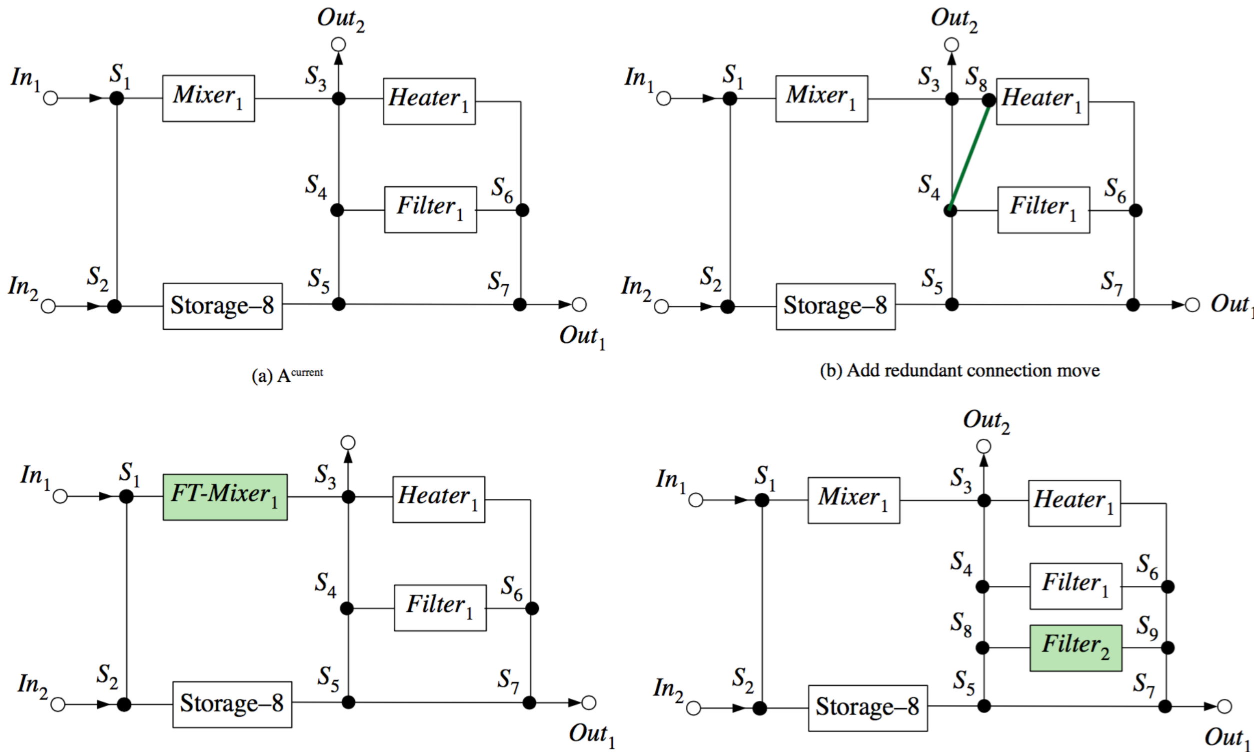
Architecture Synthesis

- Two metaheuristics:
  - Simulated Annealing
  - GRASP



Design Transformations

- Add redundant component
- Make component fault-tolerant
- Add redundant connection
- Remove redundant component
- Make component non fault-tolerant
- Remove redundant connection



(a) $A^{current}$

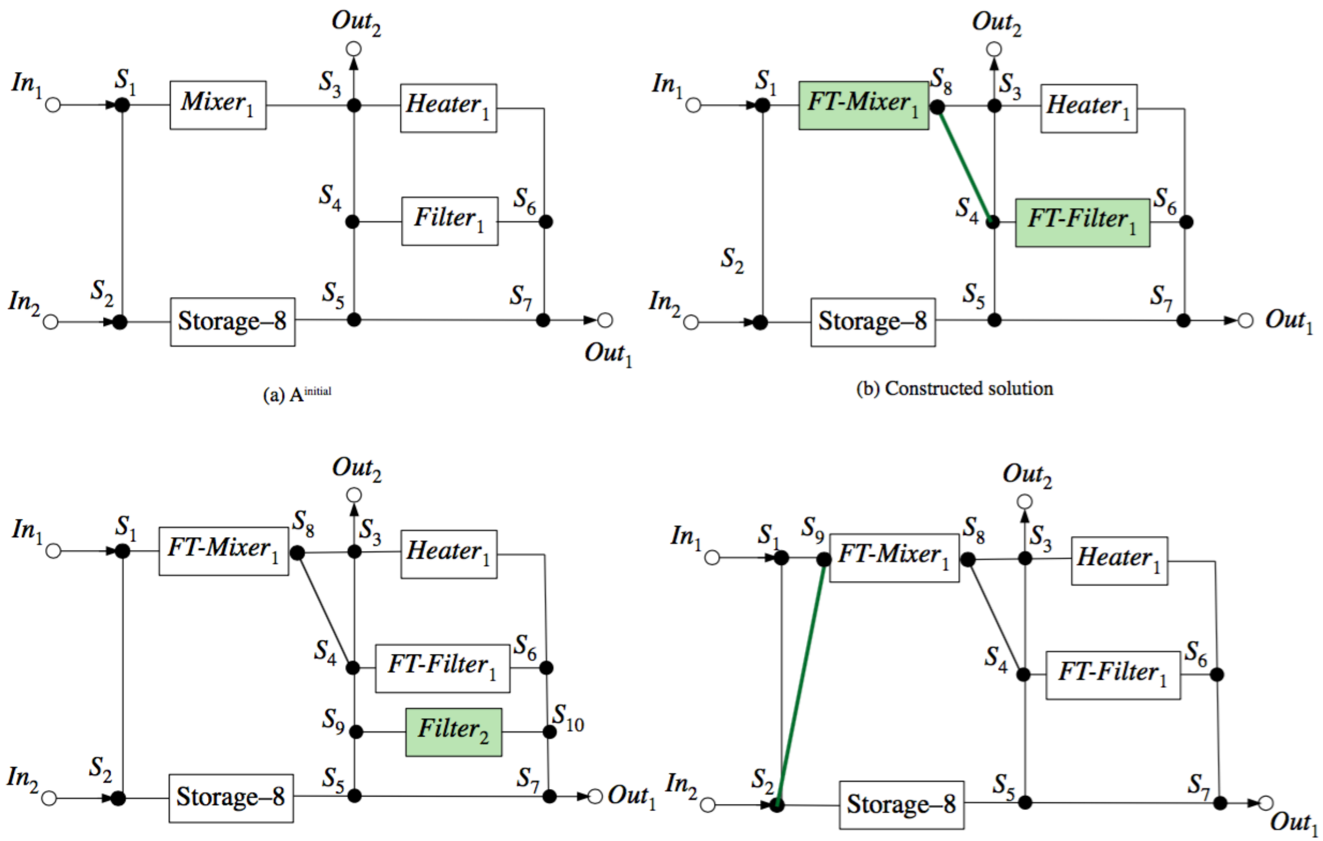
(b) Add redundant connection move

(c) Convert component to fault-tolerant move

(d) Add redundant component move

Greedily Randomized Adaptive Search Procedure

- Two phases
  - Construction phase
  - Local search phase
- Candidate list
- Ranking of components
- RCL
- Moves



(a) A^{initial}

(b) Constructed solution

(c) Local search neigbour (Add redundant Filter)

(d) Local search neighbour (Add connection between S_2 and FT-Mixer$_1$)

Methodology

Steps in the methodology:

1. Architecture design

2. Fault-tolerant architecture synthesis
   - Considering an application and a fault model

3. Physical architecture synthesis

4. Fabrication

5. Testing
   - If a fault is present and not in the fault model or it is not tolerated, discard the chip
   - Or if the application is not schedulable, the chip is discarded

Experimental Evaluation

Objective function evaluation

Initial features of benchmarks

Name	Type	$\mathcal{A}_{initial}$ $\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost	Obj_{cost}	Possible $\|\mathcal{FS}\|$	$\|\mathcal{FS}\|$	$d_{\mathcal{G}}$
S-1	Synthetic	15	17	84	22350084	121	100	50
PCR	Real-life	14	16	88	1695088	77	50	65
IVD	Real-life	52	78	274	1800274	841	100	90

The resulting netlists of the benchmarks

Name	$\mathcal{A}_{SA}$				$\mathcal{A}_{GRASP}$			
	$\|\mathcal{N}\|$	FT-$\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost	$\|\mathcal{N}\|$	FT-$\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost
S-1	24	1	35	185	15	3	20	102
PCR	19	1	27	124	14	1	17	92
IVD	53	2	80	285	52	2	78	279

Experimental Evaluation

Yield evaluation

Initial features of benchmarks

Name	Type	$\mathcal{A}_{initial}$ $\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost	Obj_{cost}	Possible $\|\mathcal{FS}\|$	$\|\mathcal{FS}\|$	$d_{\mathcal{G}}$
S-1	Synthetic	15	17	84	22350084	121	100	50
PCR	Real-life	14	16	88	1695088	77	50	65
IVD	Real-life	52	78	274	1800274	841	100	90

Only benchmark S-1 used and GRASP was the algorithm

The result of yield evaluation

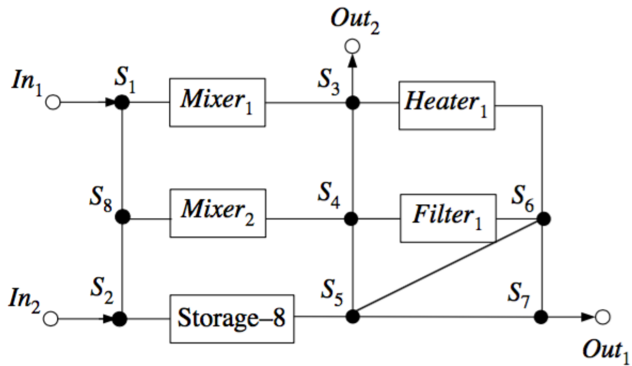
$\mathcal{A}_{initial}$ $\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost	$\|\mathcal{FS}\|$	$\mathcal{A}_{result}$ $\|\mathcal{N}\|$	FT-$\|\mathcal{N}\|$	$\|\mathcal{D}\|$	Cost	$\|FT\|$	$\|\neg FT\|$	$FT\%$
15	17	5760084	25	16	2	20	98	105	16	86.78
15	17	10540084	50	15	3	19	99	117	4	96.69
15	17	18580084	85	16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

Conclusions

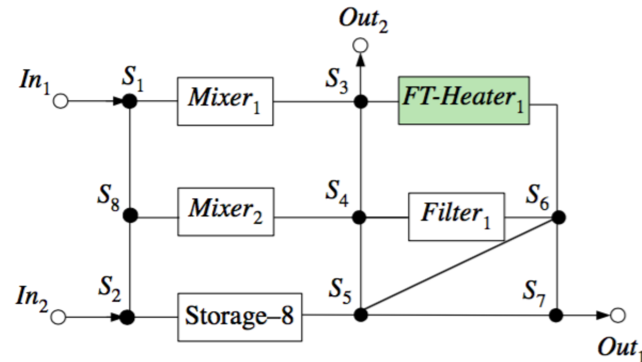
- Extended the component library with fault-tolerant components
- Proposed a fault-model for flow-based biochips
- Synthesising a fault-tolerant architecture
- Both algorithms produce fault-tolerant netlists for all benchmarks
  - GRASP provided better results in terms of solution quality and performance
- The yield evaluation proved that even generating a fraction of the possible fault scenario can lead to good fault-tolerance on all fault scenarios

Simulated Annealing

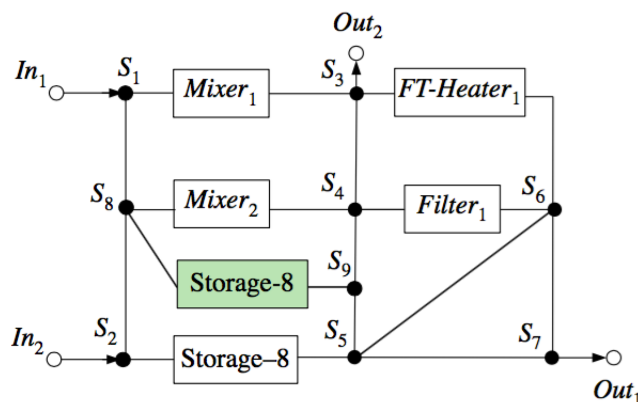
- Initial solution
- Solution acceptance
- Temperature cooling
- Moves



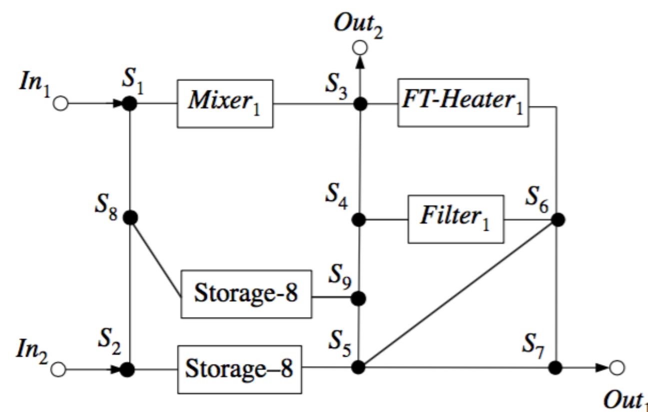
(a) $A^{current}$



(b) SA local neighbour (Make $Heater_1$ fault-tolerant)



(c) SA local neighbour (Add redundant Storage)



(d) SA local neighbour (Remove redundant $Mixer_2$)

Class Diagram