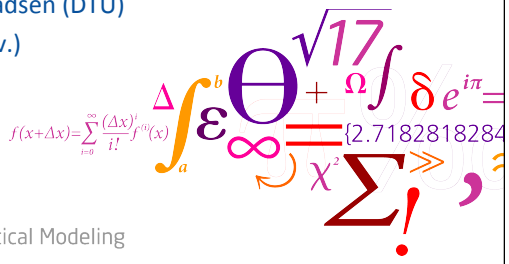
System-Level Design of Digital Microfluidic Biochips

Paul Pop
Technical University of Denmark

Acknowledgements:
Elena Maftei, Mirela Alistar, Jan Madsen (DTU)
Krishnendu Chakrabarty (Duke Univ.)

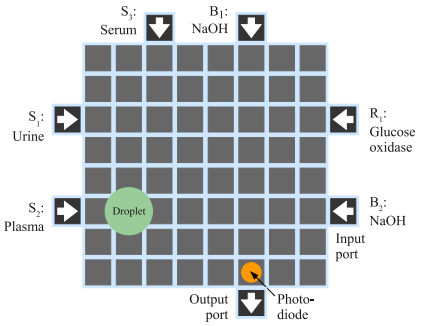


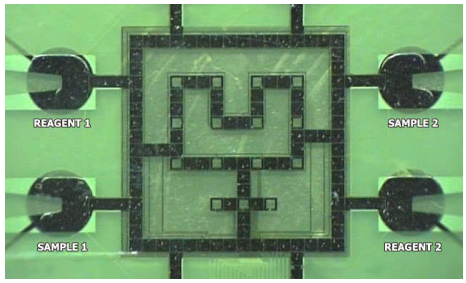
DTU Informatics
Department of Informatics and Mathematical Modeling

Outline

- Digital microfluidic biochips
  - Architecture model: module vs. routing-based
  - Application model
- Top-level design flow
  - Design tasks
- Recent research
  - Module-based synthesis
  - Routing-based synthesis
- Challenges
  - Fault-tolerant applications
  - Pin-constrained design

Architecture model

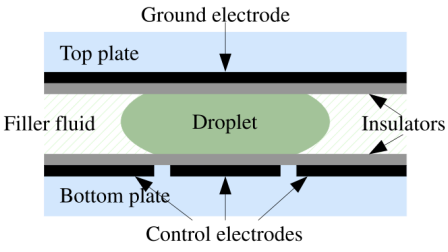




Biochip from Duke University

3

Electrowetting on Dielectric

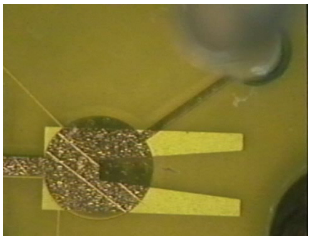


4

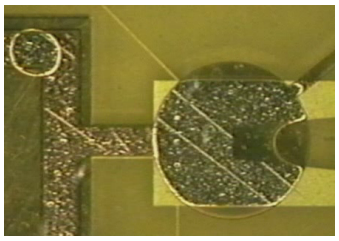
Operations, cont.

Reservoir loading

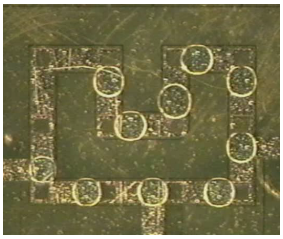
(0.1M KCL with dye)



Droplet dispensing

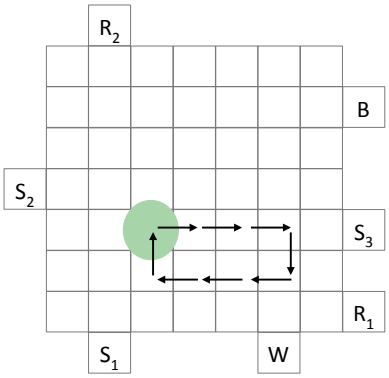


Transport on 3-phase inner bus



5

Reconfigurability



Non-reconfigurable

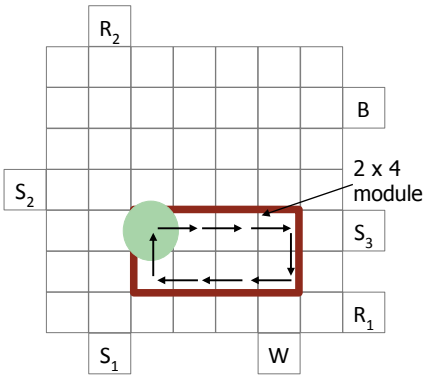
- Dispensing
- Detection

Reconfigurable

- Splitting/Merging
- Storage
- Mixing/Dilution

6

Operation execution: Module based



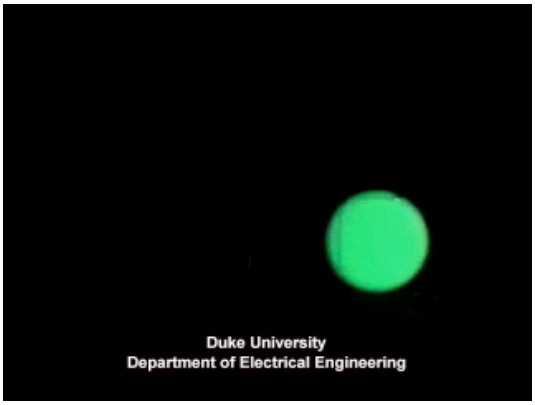
Module library

Operation	Area (cells)	Time (s)
Mix	2 x 4	3
Mix	2 x 2	4
Dilution	2 x 4	4
Dilution	2 x 2	5

7

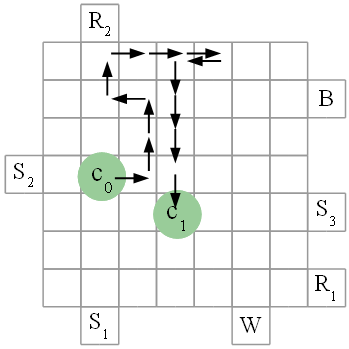
Operations: Mixing

- Droplets can move anywhere
- Fixed area: **module-based** operation execution
- Unconstrained: **routing-based** operation execution



8

Operation execution: Routing based



- Droplets can move anywhere
- Constrained to a module
  - We know the completion time from the module library.
- Unconstrained, any route
  - How can we find out the operation completion times?

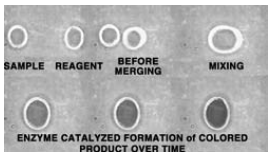
9

Application model: from this…

$$\text{Glucose} + H_2O + O_2 \xrightarrow{\text{Glucose Oxidase}} \text{Gluconic Acid} + H_2O_2$$

$$2H_2O_2 + \text{4-AAP} + \text{TOPS} \xrightarrow{\text{Peroxidase}} \text{Quinoneimine} + 4H_2O$$

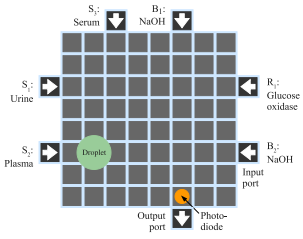
Trinder's reaction, a colorimetric enzyme-based method



Glucose assay steps on the biochip

Sample	Reagent	Enzymatic Assay
Plasma: S_1	R_1	Glucose Measurement
Serum: S_2	R_2	Lactate Measurement
Urine: S_3	R_3	Pyruvate Measurement
Saliva: S_4	R_4	Glutamate Measurement

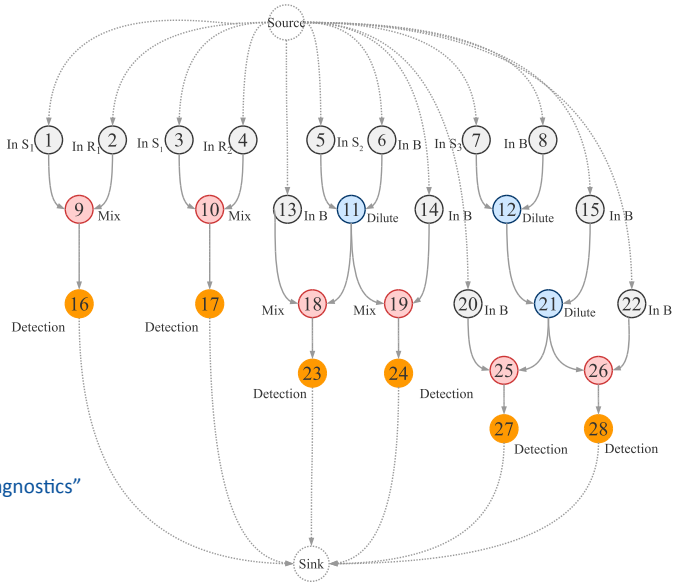
Several such reactions assays in parallel: "in-vitro diagnostics" application



Reconfigurable architecture

10

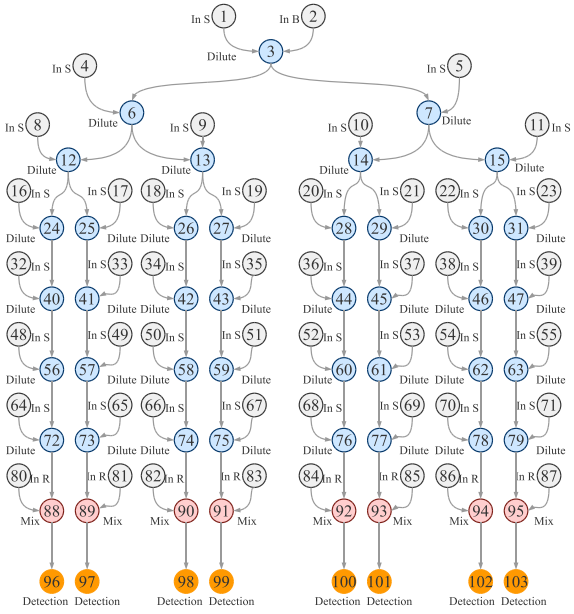
Application model: …to this—an acyclic directed graph



"in-vitro diagnostics" application

11

Another application example: "Colorimetric protein assay"



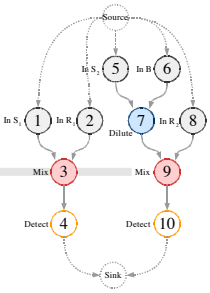
12

System-level design tasks

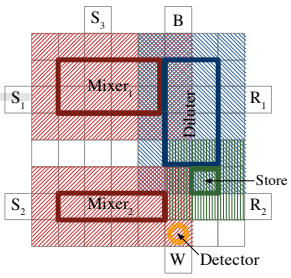
Allocation

Operation	Area (cells)	Time (s)
Mixing	2x2	6
Mixing	2x3	5
Mixing	2x4	4
Dilution	2x2	6
Dilution	2x3	5
Dilution	2x4	3
Storage	1x1	-

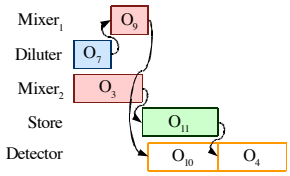
Binding



Placement & routing



Scheduling



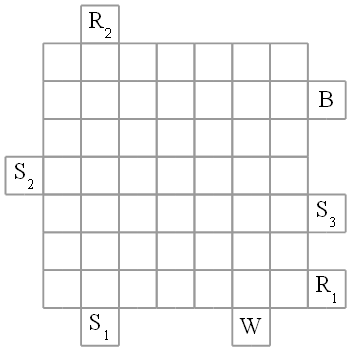
13

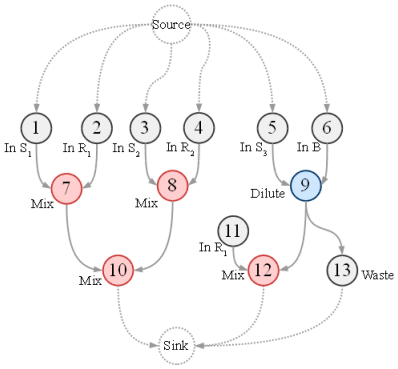
My motivation: adapt familiar design methods to a new area

	FPGA	Digital biochip
Basic Devices	Transistors	Control electrodes
	Net Wires	Reservoirs
	Clock lines	Transparent cells
Tiles	RAM	Mixers
	Multiplexer	Transport bus
	CLBs	Optical detectors
Systems	Configured FPGA	Configured biochip

14

Module-Based Synthesis

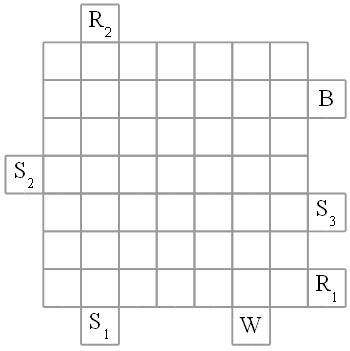


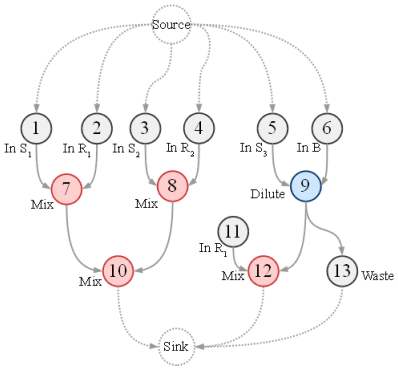


15

Module-Based Synthesis

O7
O8
O9

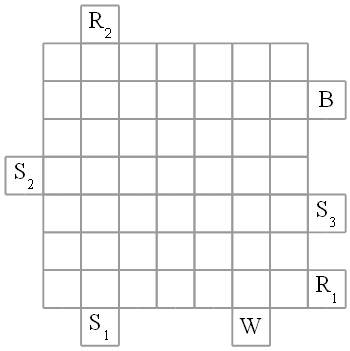


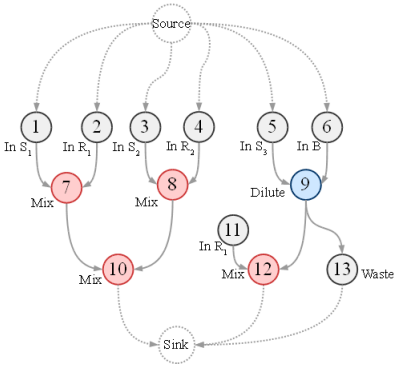


16

Module-Based Synthesis

O7
O8
O9

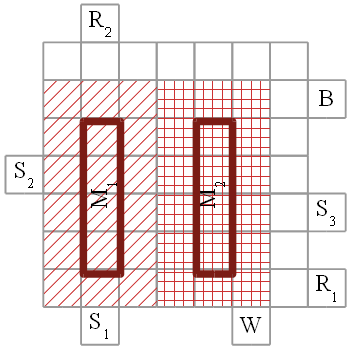


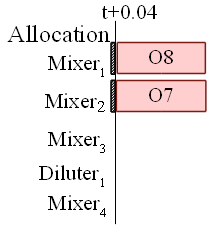


17

Module-Based Synthesis

O7 1x4
O8 1x4

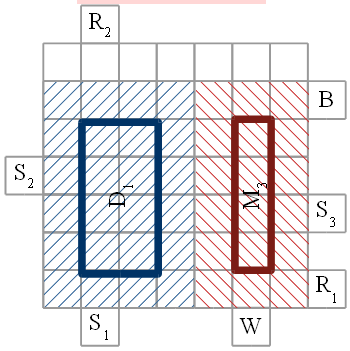


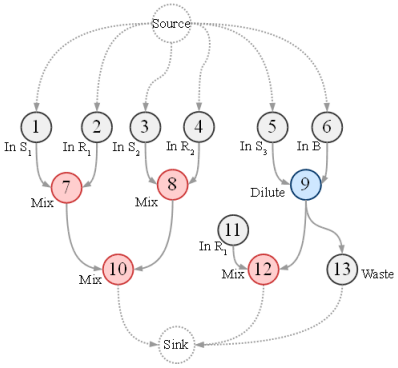


18

Module-Based Synthesis

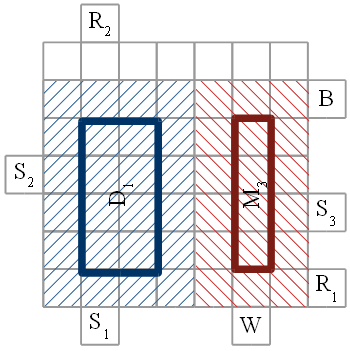
O9	2x4
O10	1x4

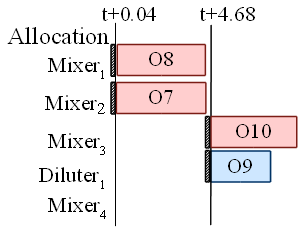




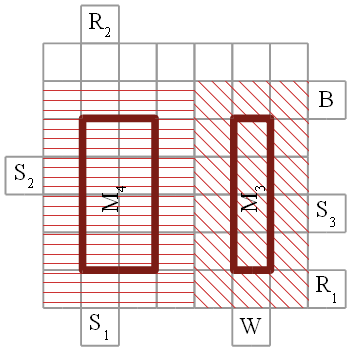
Module-Based Synthesis

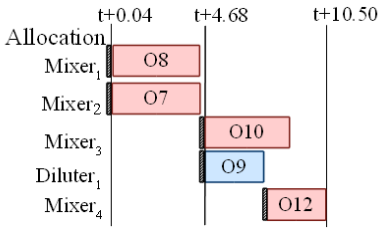
O9	2x4
O10	1x4





Module-Based Synthesis





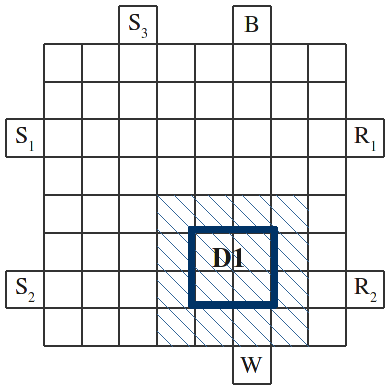
21

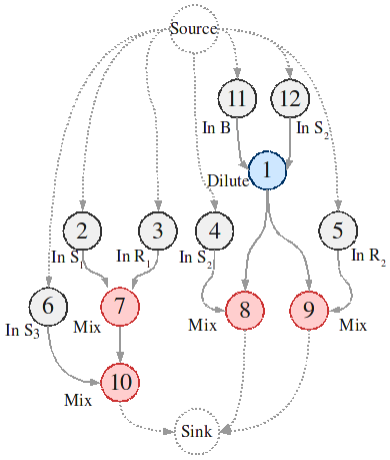
Problem Formulation

- **Given**
  - Application: graph
  - Biochip: array of electrodes
  - Library of modules
- **Determine**
  - **Allocation** of modules from modules library
  - **Binding** of modules to operations in the graph
  - **Scheduling** of operations
  - **Placement** of modules on the array
- **Such that**
  - the application execution time is minimized

22

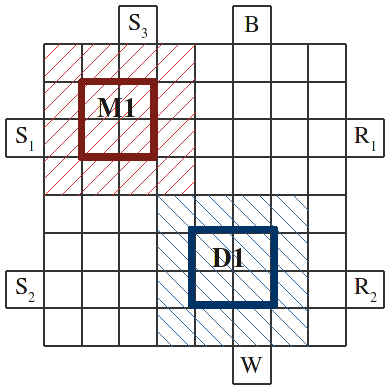
Reconfigurability

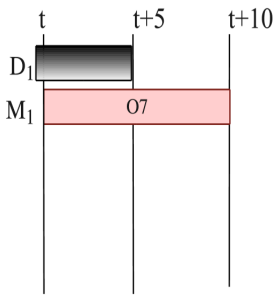




23

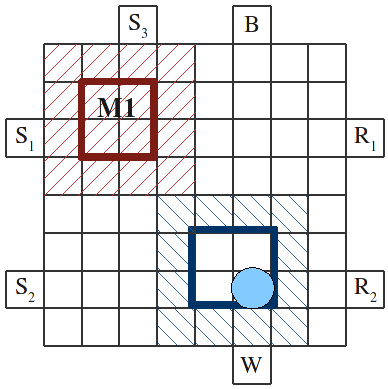
Reconfigurability

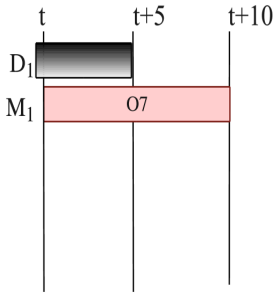




24

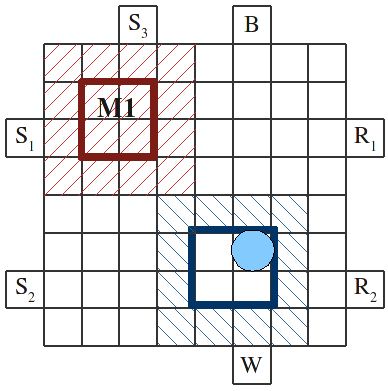
Reconfigurability

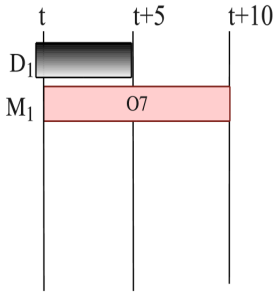




25

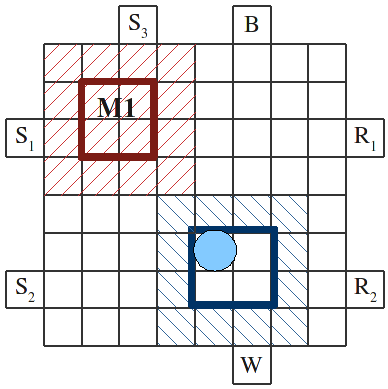
Reconfigurability

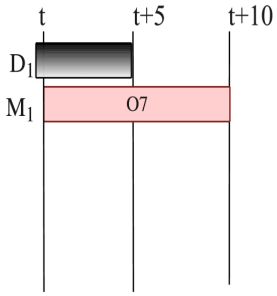




26

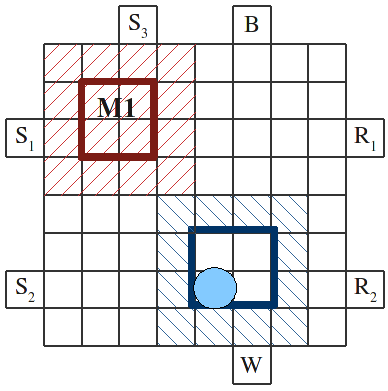
Reconfigurability

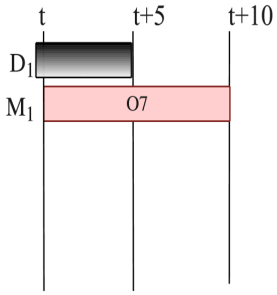




27

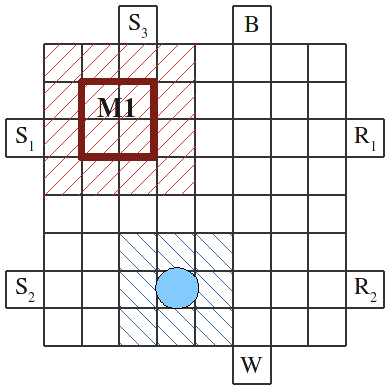
Reconfigurability

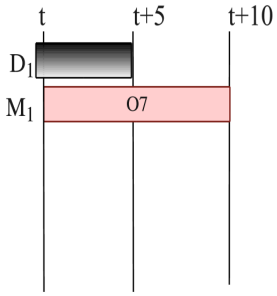




28

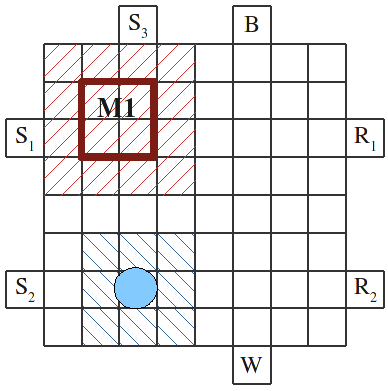
Reconfigurability

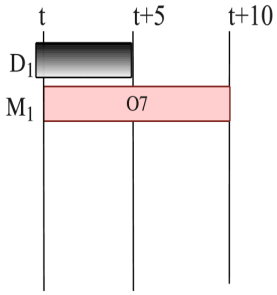




29

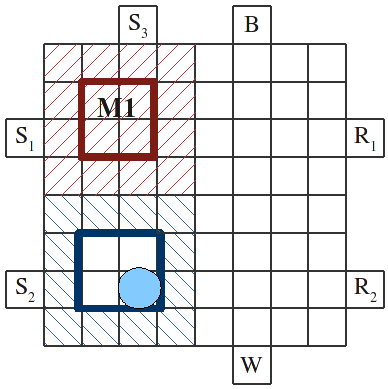
Reconfigurability

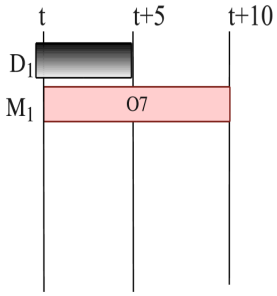




30

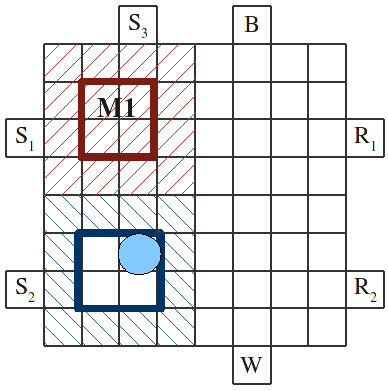
Reconfigurability

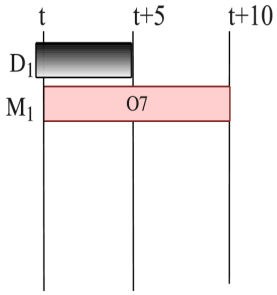




31

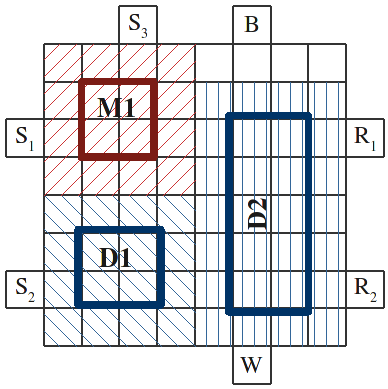
Reconfigurability

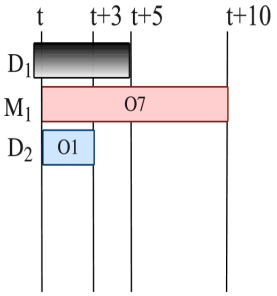




32

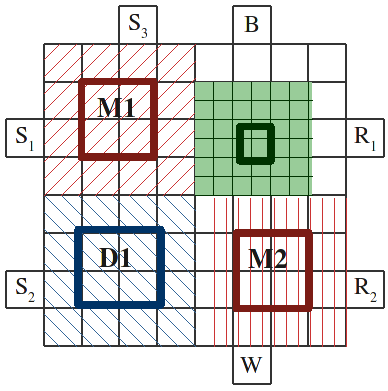
Reconfigurability

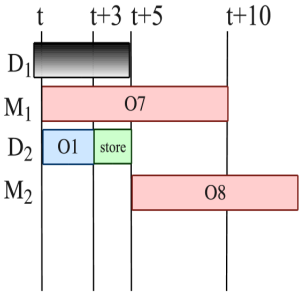




33

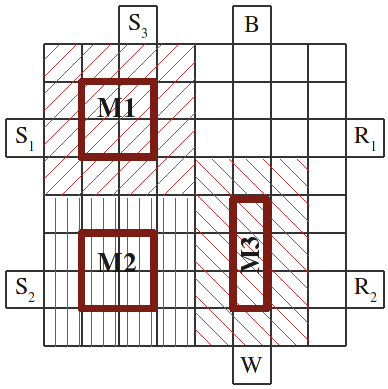
Reconfigurability

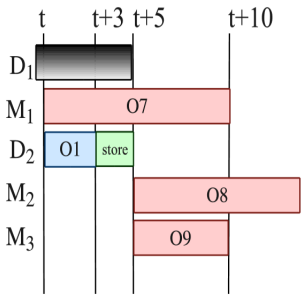




34

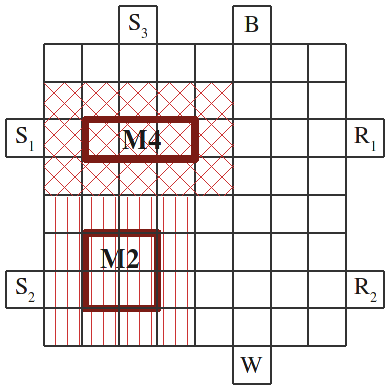
Reconfigurability

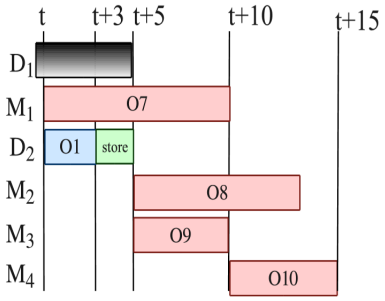




35

Reconfigurability

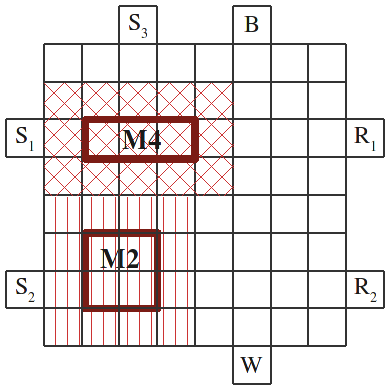


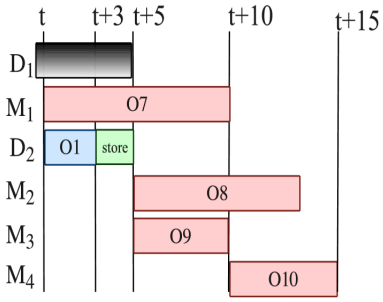


36

Reconfigurability

Without dynamic reconfiguration: t+18





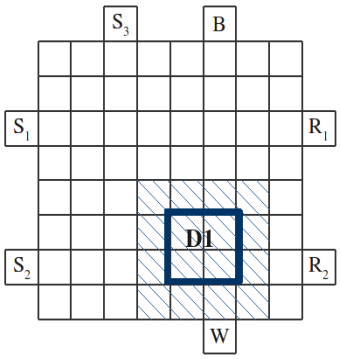
37

Solution

- Binding of modules to operations — **Tabu Search**
- Schedule of the operations — **List Scheduling**
  - Placement of modules performed inside scheduling
- Placement of the modules — **Maximal Empty Rectangles**
  - Free space manager based on [Bazargan et al. 2000] that divides free space on the chip into overlapping rectangles
- Other solutions proposed in the literature:
  - Integer Linear Programming
  - Simulated Annealing

38

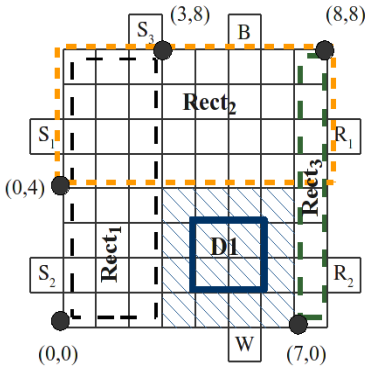
Dynamic Placement Algorithm



Operation	Module
O_7 (mix)	M_1 (2x2)
O_1 (diluter)	D_2 (2x5)

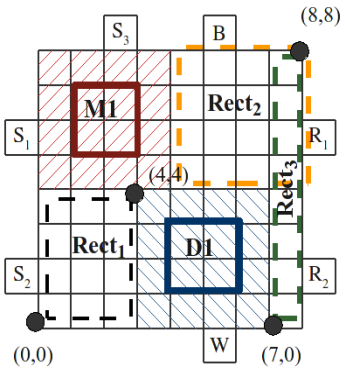
39

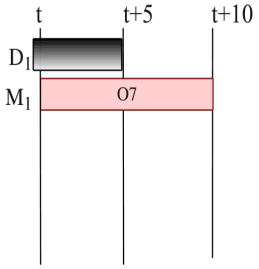
Dynamic Placement Algorithm



40

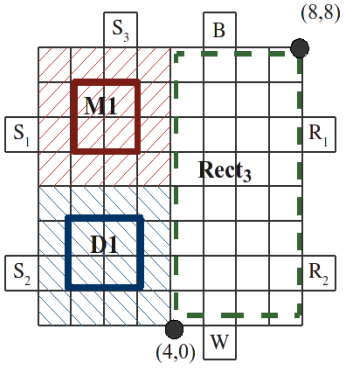
Dynamic Placement Algorithm

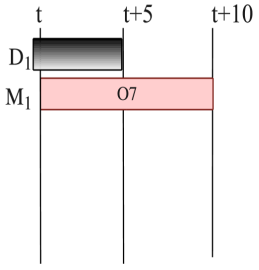




41

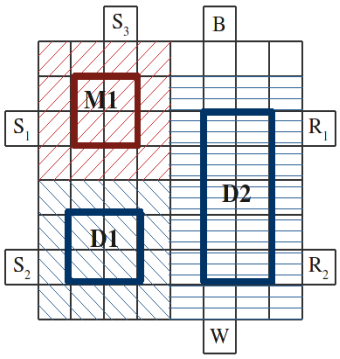
Dynamic Placement Algorithm

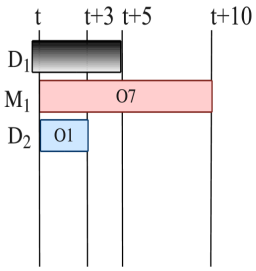




42

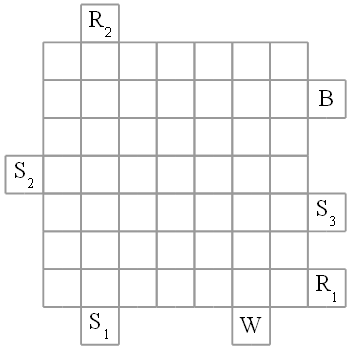
Dynamic Placement Algorithm

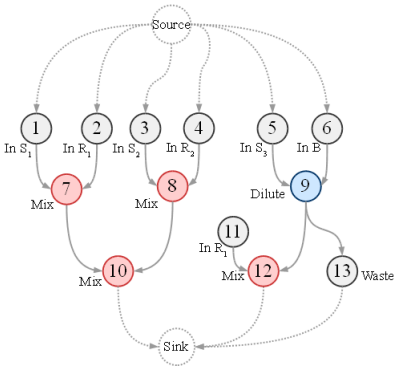




43

Routing-Based Synthesis

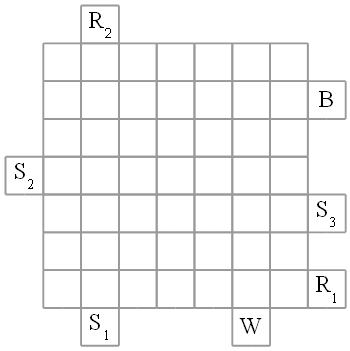


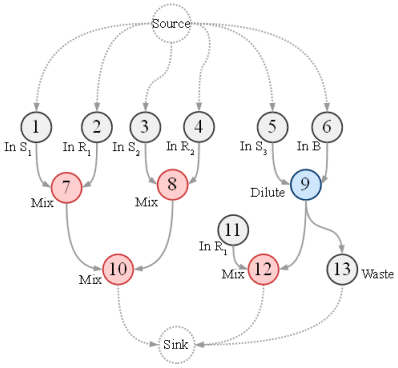


44

Routing-Based Synthesis

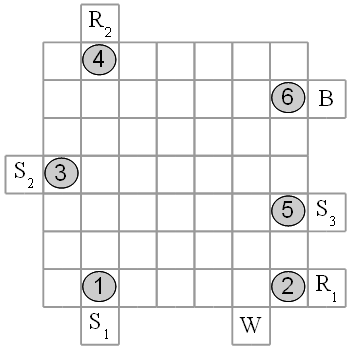
O7
O8
O9

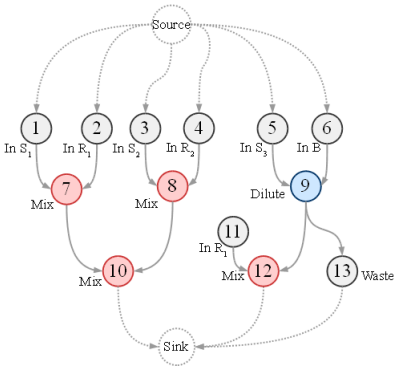




45

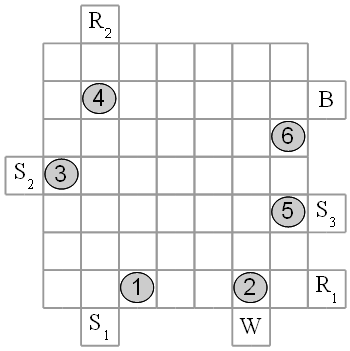
Routing-Based Synthesis

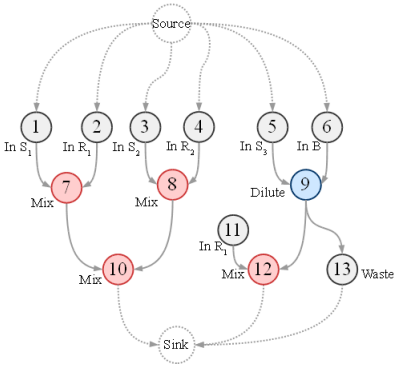




46

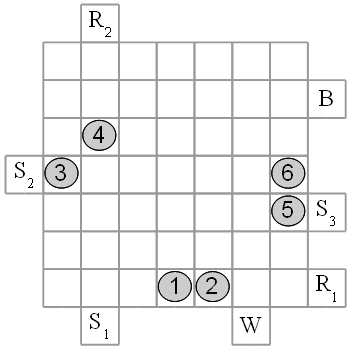
Routing-Based Synthesis

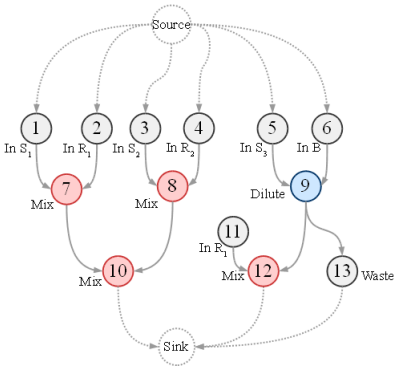




47

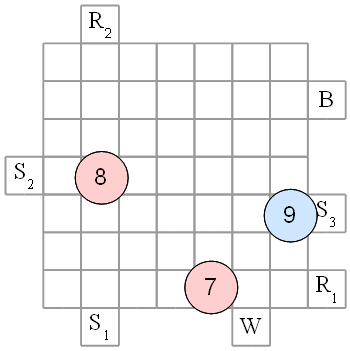
Routing-Based Synthesis

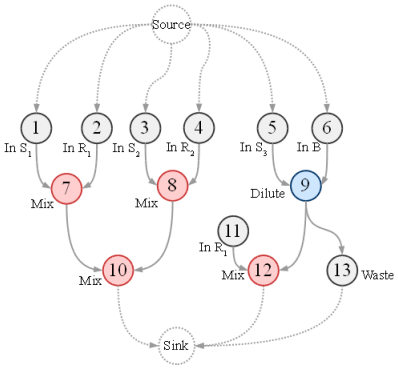




48

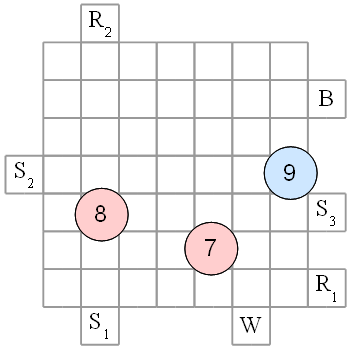
Routing-Based Synthesis

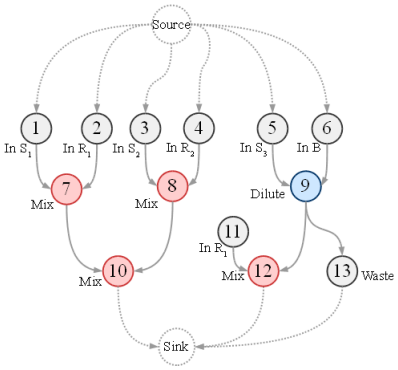




49

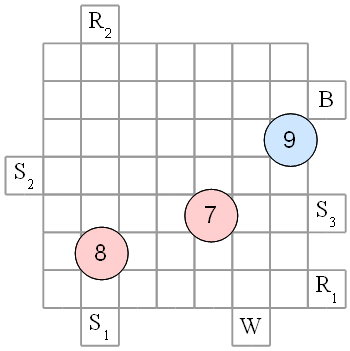
Routing-Based Synthesis

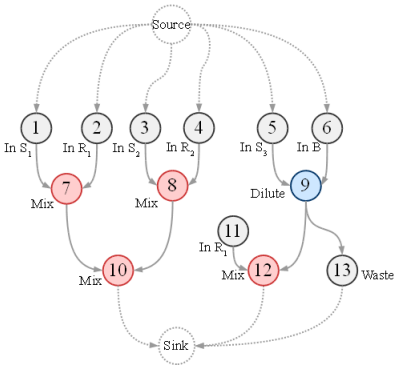




50

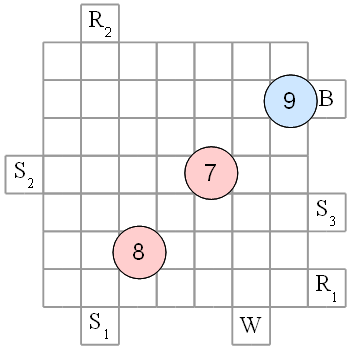
Routing-Based Synthesis

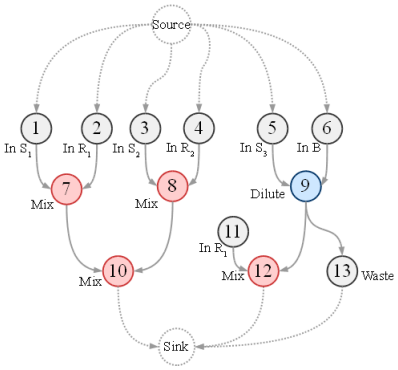




51

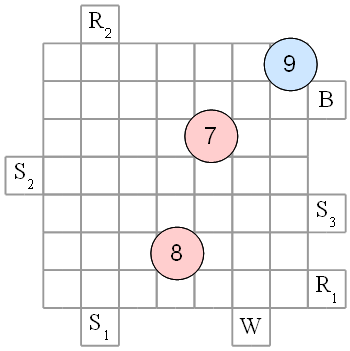
Routing-Based Synthesis

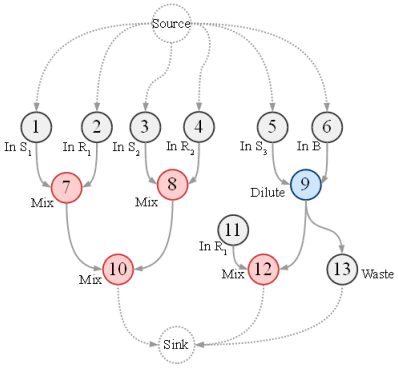




52

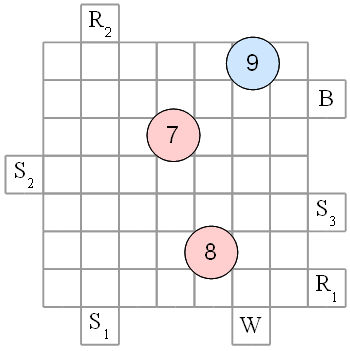
Routing-Based Synthesis

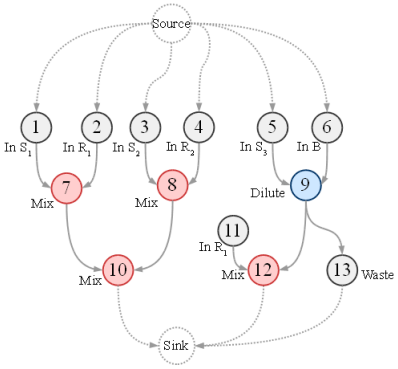




53

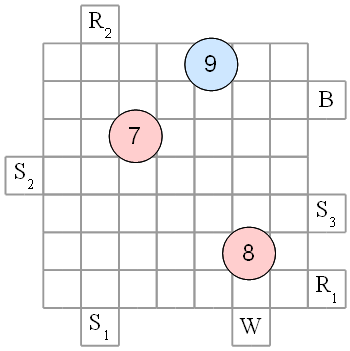
Routing-Based Synthesis

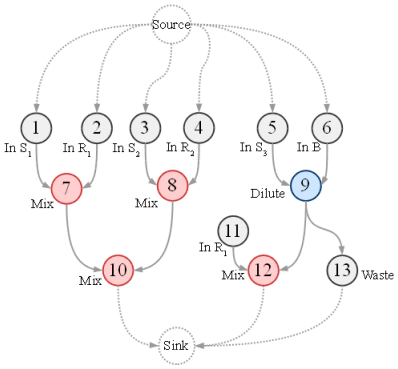




54

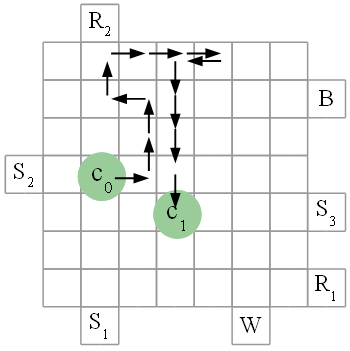
Routing-Based Synthesis





55

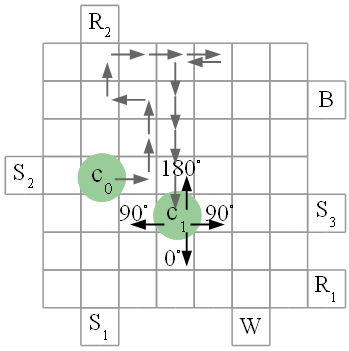
When will the operations complete?



- For module-based synthesis we know the *completion time* from the module library.
- But now there are no modules, the droplets can move anywhere:
  - How can we find out the operation *completion times*?

56

Characterizing operations



- If the droplet does not move: very slow mixing by diffusion
- If the droplet moves, how long does it take to complete?
- Mixing percentages:

p^0, p^{90}, p^{180} ?

57

Characterizing operations

Operation	Area(cells)	Time(s)
Mix/Dlt	2x4	2.8
Mix/Dlt	1x4	4.6
Mix/Dlt	2x3	5.6
Mix/Dlt	2x2	9.96

- We know how long an operation takes on modules
- Starting from this, can determine the percentages?

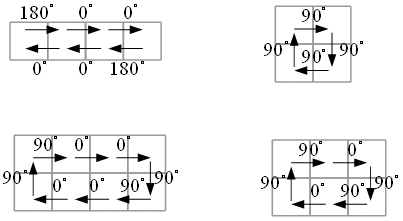
58

Decomposing modules

Safe, conservative estimates

p^{90} = 0.1%, p^{180} = -0.5%,
p^{0} = 0.29% and 0.58%

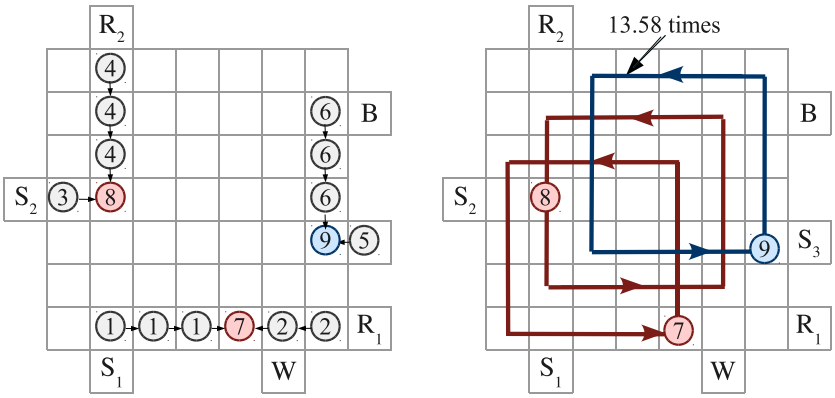
Operation	Area(cells)	Time(s)
Mix/Dlt	2x4	2.8
Mix/Dlt	1x4	4.6
Mix/Dlt	2x3	5.6
Mix/Dlt	2x2	9.96



Moving a droplet one cell takes 0.01 s.

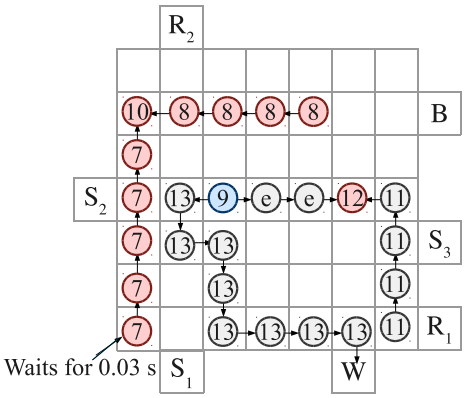
59

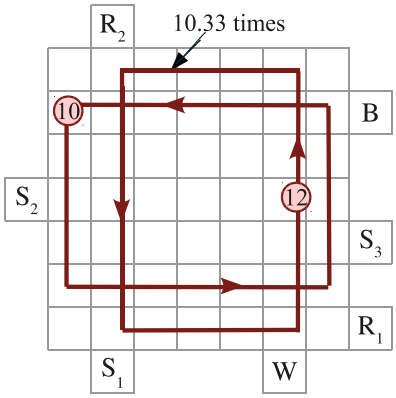
Routing-Based Synthesis



60

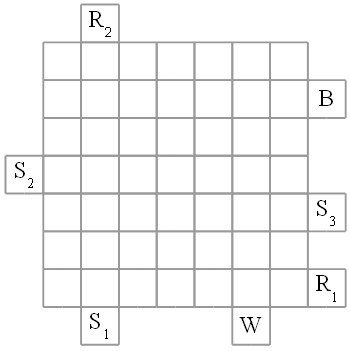
Routing-Based Synthesis

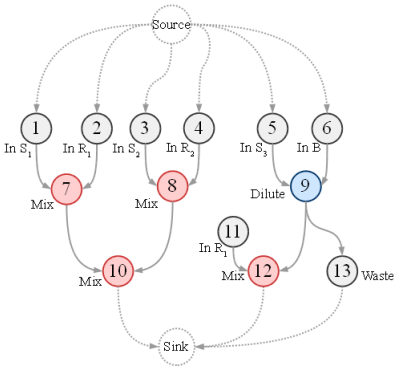




61

Routing- vs. Module-Based Synthesis

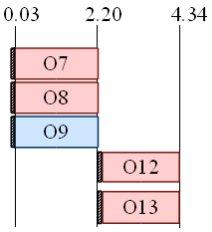




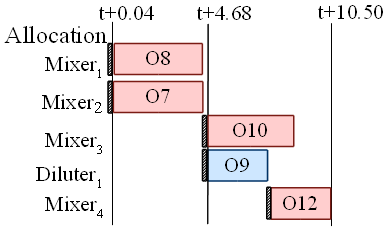
62

Routing- vs. Module-Based Synthesis

Routing-Based Synthesis



Module-Based Synthesis



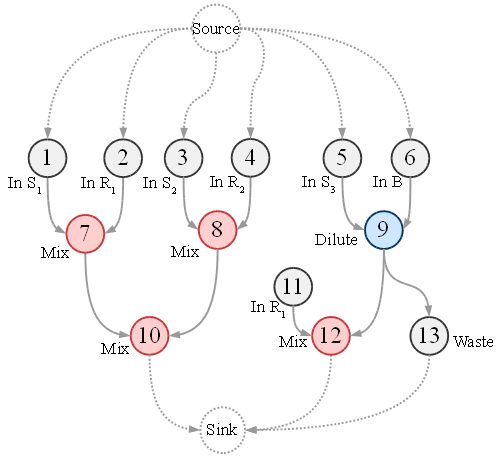
63

Problem Formulation

- **Given**
  - Application: graph
  - Biochip: array of electrodes
  - Library of non-reconfigurable devices
- **Determine**
  - **Droplet routes** for all reconfigurable operations
  - **Allocation** and **binding** of non-reconfigurable modules from a library
  - **Scheduling** of operations
- **Such that**
  - the application completion time is minimized

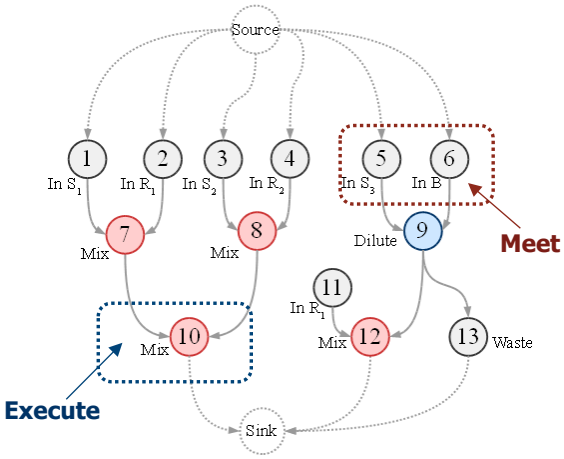
64

Proposed Solution



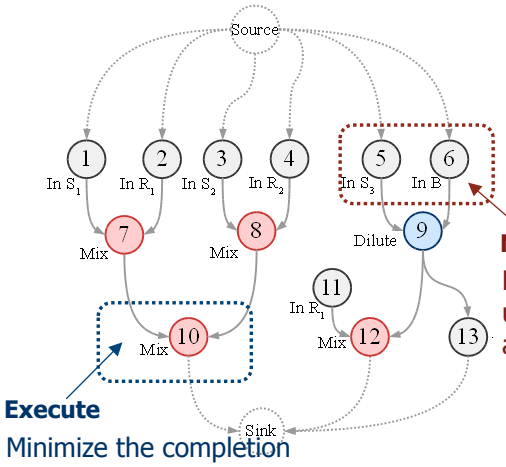
65

Proposed Solution



66

Proposed Solution



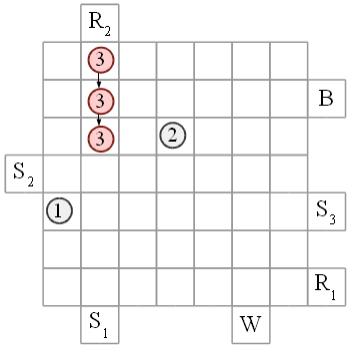
Meet
Minimize the time until the droplet(s) arrive at destination

Execute
Minimize the completion time for the operation

67

GRASP-Based Heuristic

- **G**reedy **R**andomized **A**daptive **S**earch **P**rocedure

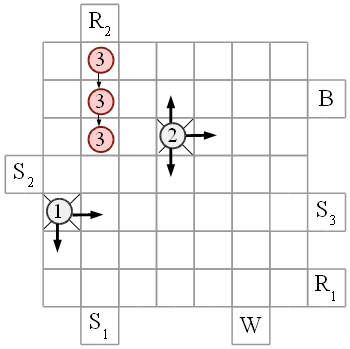


- For each droplet:
  - Determine possible moves
  - Evaluate possible moves
  - Make a list of best N possible moves
  - Perform a randomly chosen possible move

68

GRASP-Based Heuristic

- Greedy Randomized Adaptive Search Procedure

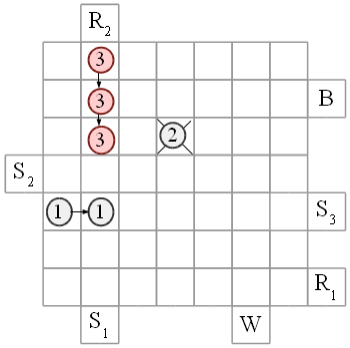


- For each droplet:
  - Determine possible moves
  - Evaluate possible moves
  - Make a list of best N possible moves
  - Perform a randomly chosen possible move

69

GRASP-Based Heuristic

- Greedy Randomized Adaptive Search Procedure

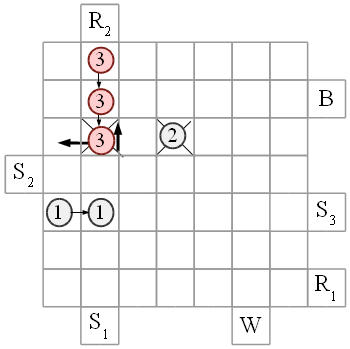


- For each droplet:
  - Determine possible moves
  - Evaluate possible moves
  - Make a list of best N possible moves
  - Perform a randomly chosen possible move

70

GRASP-Based Heuristic

- Greedy Randomized Adaptive Search Procedure

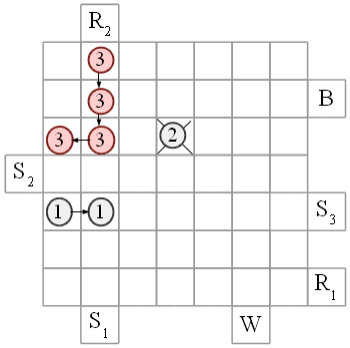


- For each droplet:
  - Determine possible moves
  - Evaluate possible moves
  - Make a list of best N possible moves
  - Perform a randomly chosen possible move

71

GRASP-Based Heuristic

- Greedy Randomized Adaptive Search Procedure



- For each droplet:
  - Determine possible moves
  - Evaluate possible moves
  - Make a list of best N possible moves
  - Perform a randomly chosen possible move

72

Experimental Evaluation

Routing-Based Synthesis (**RBS**) vs. to Module-Based Synthesis (**MBS**)

Application	Area	Best	
		RBS	MBS
In-vitro (28 operations)	8 × 9	68.43	72.94
	8 × 8	68.87	82.12
	7 × 8	69.12	87.33
Proteins (103 operations)	11 × 11	113.63	184.06
	11×10	114.33	185.91
	10 × 10	115.65	208.90

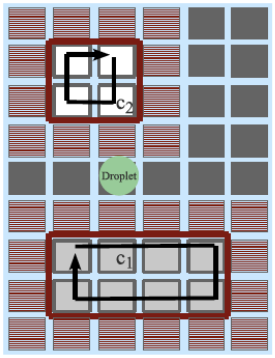
73

Discussion

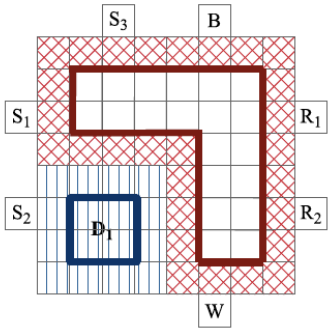
- **Module**-based vs. **routing**-based
  - Module-based needs an extra routing step between the modules; Routing-based performs unified synthesis and routing
  - Module-based wastes space: only one module-cell is used; Routing-based exploits better the application parallelism
  - Module-based can contain the contamination to a fixed area;
  - We have extended routing-based to address contamination
- **Hybrid** approaches are also possible
  - Non-rectangular modules
  - Droplet-aware module-based synthesis
  - Area-constrained routing-based synthesis

74

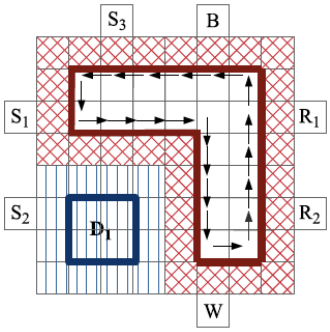
Non-rectangular modules



(a) Mixing operation



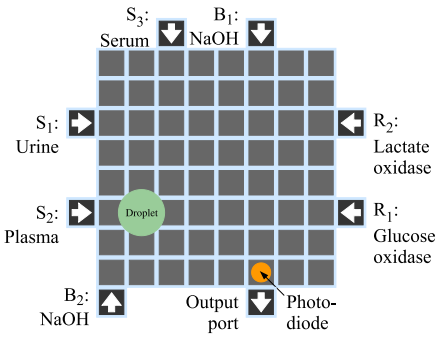
(b) L-shaped mixer



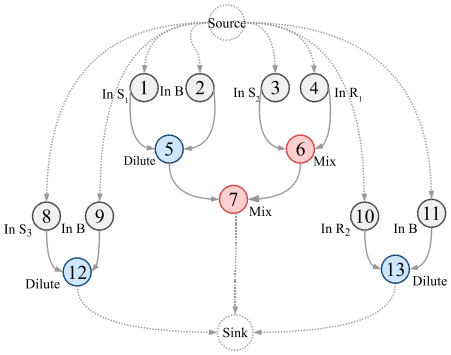
(c) Movement pattern

75

Droplet-Aware Operation Execution



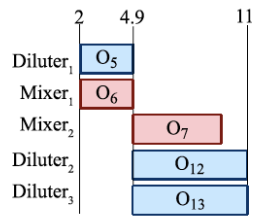
(a) Microfluidic array



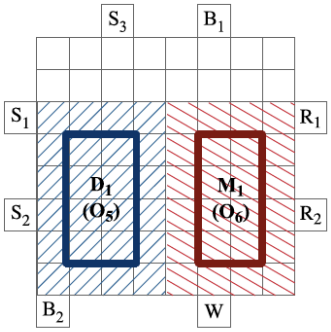
(b) Application graph

76

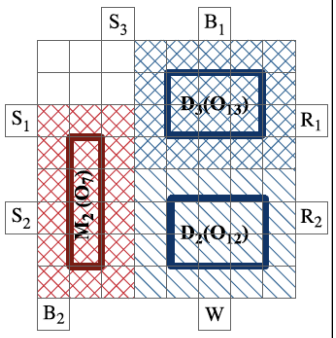
Droplet-Aware Operation Execution



(a) Schedule with segregation cells



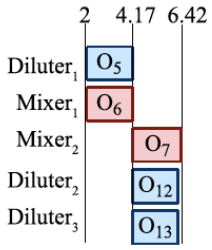
(b) Placement at $t = 2$



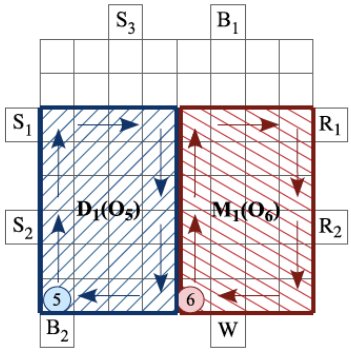
(c) $t = 4.9$

77

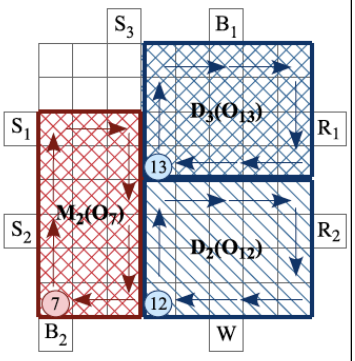
Droplet-Aware Operation Execution



(a) Improved schedule



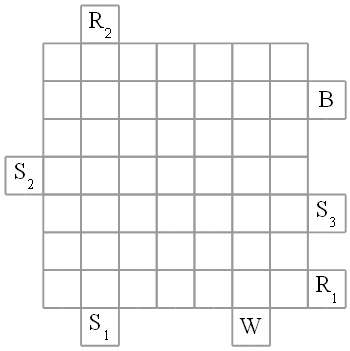
(b) Placement at $t = 2$

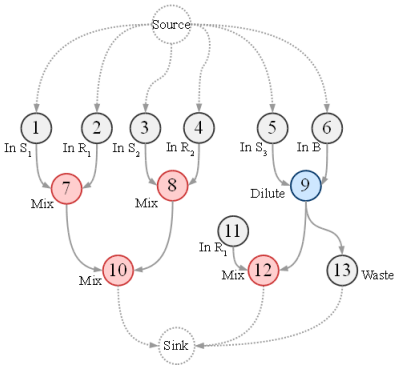


(c) $t = 4.17$

78

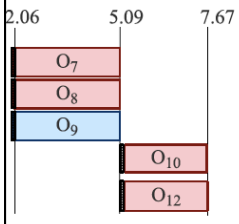
Area-constrained routing-based synthesis



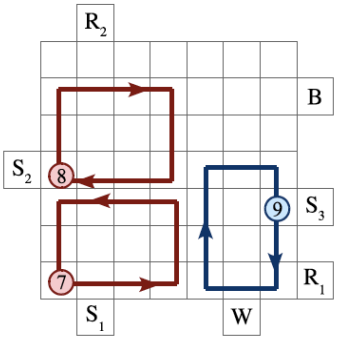


79

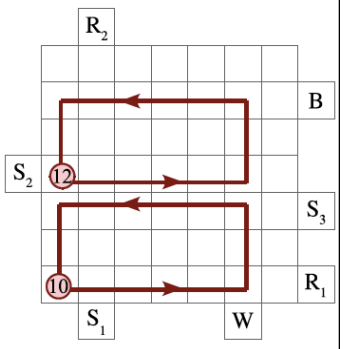
Area-constrained routing-based sysnthesis



(a) Schedule



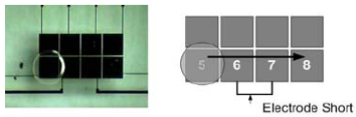
(b) $t \in (2.06, 5.09]$



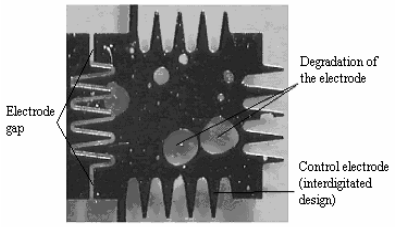
(c) $t \in (5.18, 7.67]$

80

Synthesis Challenges: Faults

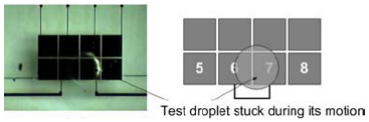


Electrode short



Electrode degradation

Imperfect splitting



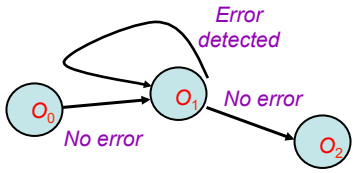
Hindered transportation

81

Motivation for Error Recovery

- Verify correctness of fluidic operations in bioassay
  - Monitor bioassay status to find errors
  - Parameters for monitoring: volume of product droplet, sample concentration, *others*?
- Correct errors as soon as possible
  - Re-execute only the erroneous part of bioassay
- Drawback of current synthesis tools
  - Only provide a "data path", no control or feedback mechanism
  - Monitor bioassay result at the end and re-execute the entire assay to correct errors

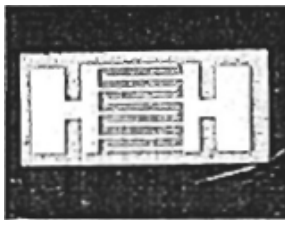
Need control-path design for error detection and recovery



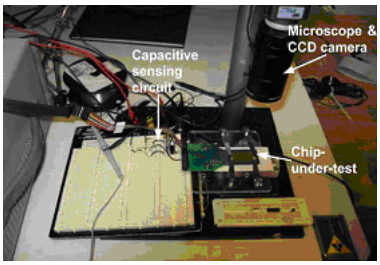
82

Droplet Detection Mechanisms

- Capacitive-sensing circuit for volumetric test
- Optical detection for concentration test



Thin-film MSM detector (S.-W. Seo, PhD Thesis 2003)



Capacitive-sensing circuit
(M. G. Pollack, PhD Thesis 2001)

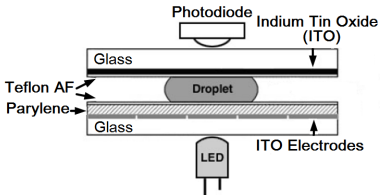
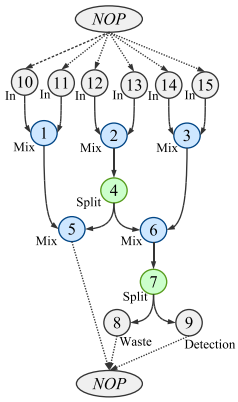
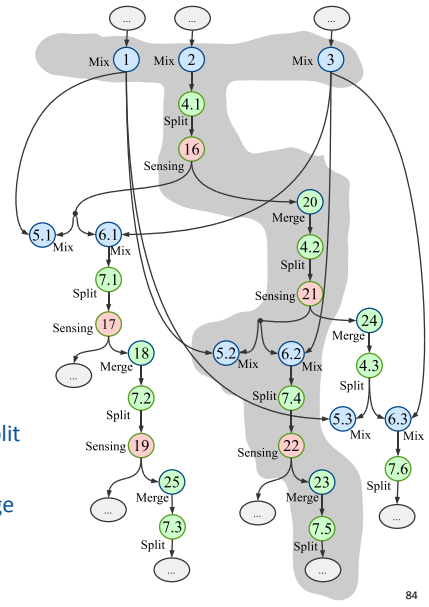


Photo-diode detector (Srinivasan et al., MicroTAS'03)

83

Fault-tolerant graph: captures fault scenarios due to split operations



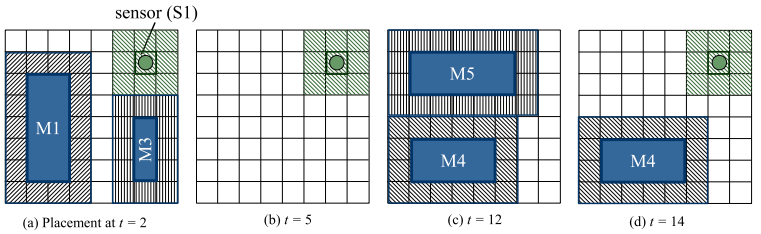


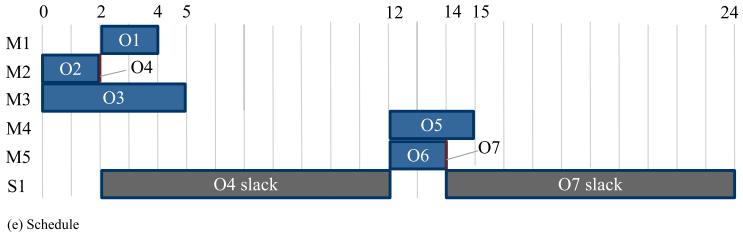
- A sensing operation is introduced after each split
- If the split was OK, the graph continues
- If the split was NOT OK, we retry: insert a merge operation followed by another split

Assumption: at most two consecutive errors

84

Straightforward scheduling

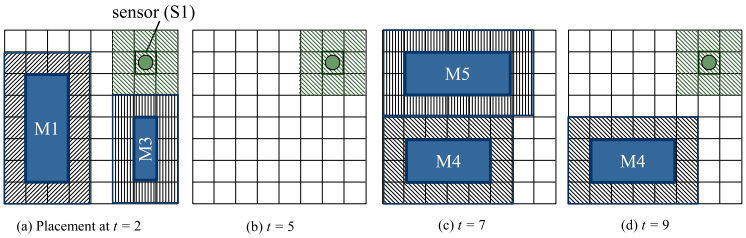


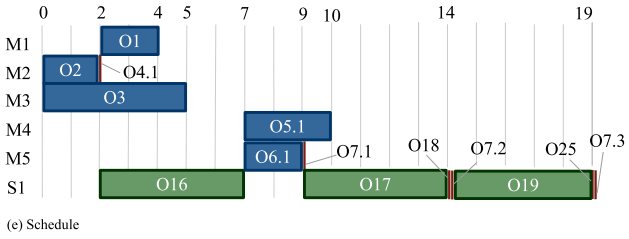


Adding worst-case slack after each split to allow for recovery

85

Scheduling the fault-tolerant graph: backup schedules for fault scenarios

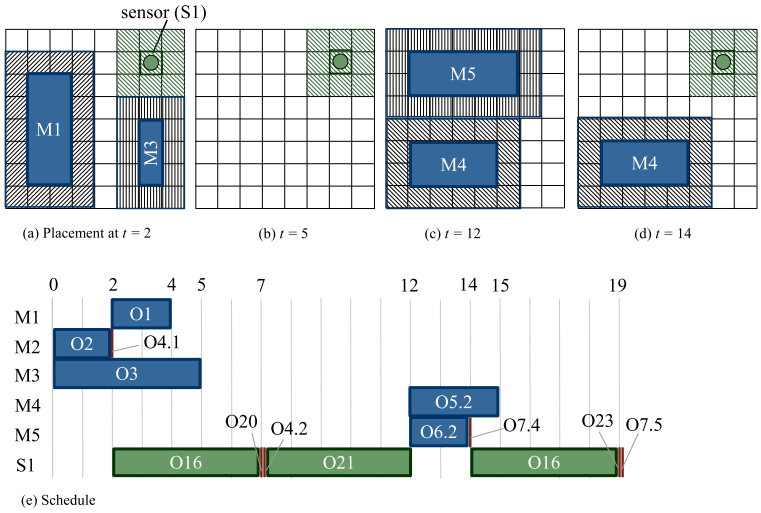




Fault-tolerant schedule for two faults in O7

86

Scheduling the fault-tolerant graph: backup schedules for fault scenarios

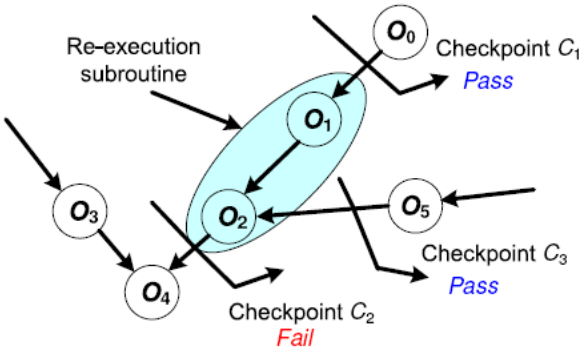


Fault-tolerant schedule for faults in O4 and O7

87

Another approach: Control-Path Design

- Add *checkpoints* to monitor outcomes of fluidic operations
  - Checkpoint: storage of the intermediate product droplet
  - Add checkpoints based on error-propagation estimates
- Assign each checkpoint a *re-execution subroutine*
  - Subroutine: fluidic operations between checkpoints
  - Correct the detected error by re-executing the subroutine

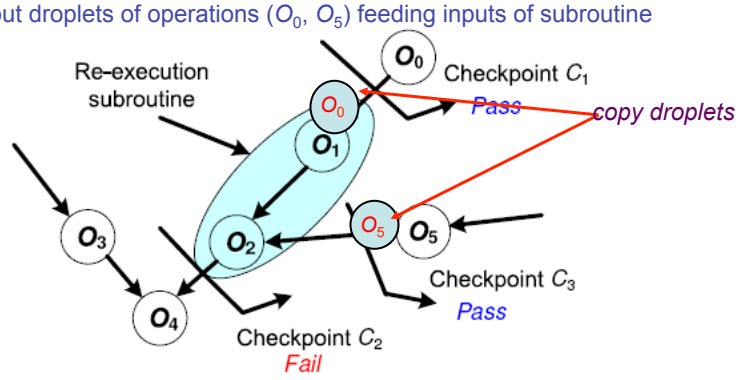


- Status at checkpoints
  - C_1: Pass
  - C_2: Fail
  - C_3: Pass
- Re-execution subroutine for C_2
  - Operations O_1 and O_2

88

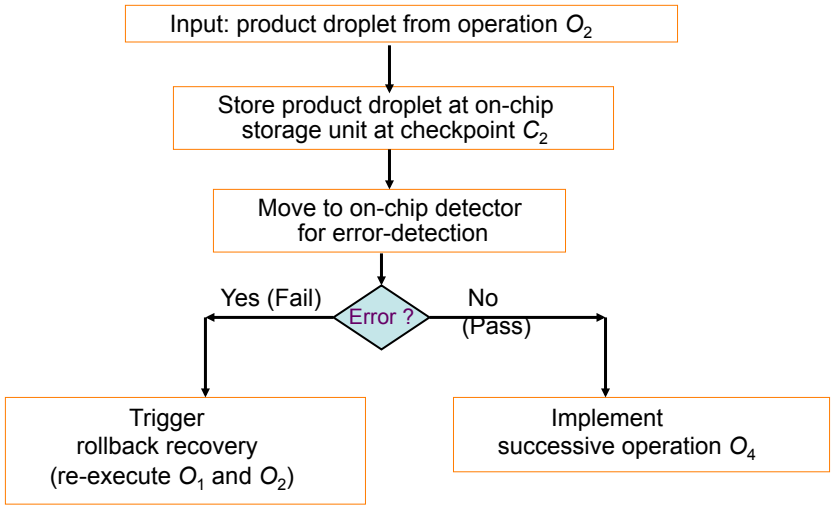
Control-Path Design

- Error detection at the checkpoint
  - Performed for intermediate product droplet at the checkpoint
  - Concentration test (using photo-detector)
  - Volumetric test (using capacitive-sensing circuit)
- Droplet preparation for re-execution subroutine
  - Copy droplets are consumed during re-execution of a subroutine
  - Output droplets of operations (O_0, O_5) feeding inputs of subroutine



Control-Path Design

- Implementation flow for error recovery at checkpoint C_2



90

90

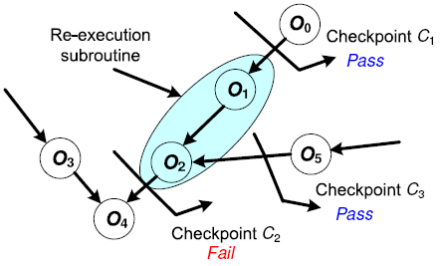
Time Cost for Control Path Design

- Part 1: time cost for the storage of the intermediate product droplet at the checkpoint (can be omitted)
- Part 2: time cost for transporting the intermediate product droplet to an on-chip detector (can be omitted)
- Part 3: time cost for error-detection
  - Typically 5 seconds for an LED-photodiode detector
  - Capacitive-sensing circuit operates at relatively high frequency (15 kHz)
- Part 4: time cost for implementing the re-execution subroutine
  - Sub-part 1: time cost for retrieving stored copy droplets and bringing to inputs of fluidic operations in the subroutine
  - Sub-part 2: time cost for re-executing the subroutine (e.g., operations O_1 and O_2 for checkpoint C_2)

91

Area Cost of Control Path Design

- Type 1: reconfigurable fluidic devices (no space cost)
  - Include storage units for the product droplets and copy droplets
  - Dynamically created using available electrodes
- Type 2: non-reconfigurable fluidic devices
  - Mainly include photo-detectors
  - Space cost: a photo-detector occupies one electrode, and the adjacent eight electrodes are used as the guard ring



- Type 1: storage units
  - Intermediate droplets for O_0, O_2 and O_5
  - Copy droplets for O_0 and O_5
- Type 2: photo-detectors
  - For checkpoints C_1, C_2 and C_3

92

92

Error-Propagation Estimates

- Add checkpoints based on error-propagation estimates
  - Add a checkpoint when output error-limit exceeds $E_{threshold}$
- Dispensing operation (Intrinsic error limit: E_{Ds})
  - Output volume: $(1 \pm E_{Ds})\mathrm{x}$
- Transportation (Intrinsic error limit: E_{Tran})
  - Volume loss due to adsorption at electrode surface
  - Input volume: $(1 \pm I)\mathrm{x}$
  - Output volume: $(1 \pm \sqrt{I^2 + E^2_{Tran}})\mathrm{x}$

Assumption: Intrinsic errors are independent Gaussian random variables

J. R. Taylor, *An Introduction to Error Analysis: the Study of Uncertainties of Physical Measurements*, 1982

93

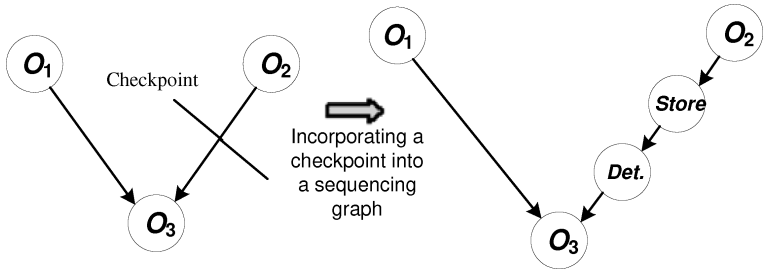
Error-Propagation Estimates

- Mix operation (Intrinsic error limit: E_{Mix})
  - Volume loss due to evaporation and adsorption
  - Input volume: $(1 \pm I_1)\mathrm{x}$ and $(1 \pm I_2)\mathrm{x}$
  - Output volume: $(1 \pm \sqrt{(0.5I_1)^2 + (0.5I_2)^2 + E^2_{Mix}})\mathrm{x}$
- Split operation (Intrinsic error limit: E_{Slt})
  - Unevenly split due to difference in applied voltages
  - Input volume: $(1 \pm I)\mathrm{x}$
  - Output volume: $(1 \pm \sqrt{I^2 + (2E_{Slt})^2})\mathrm{x}$
- Dilution (Intrinsic error limit: E_{Dlt})
  - Output volume: $(1 \pm \sqrt{(0.5I_1)^2 + (0.5I_2)^2 + (2E_{Dlt})^2})\mathrm{x}$

94

Control-Path Synthesis

- Map a checkpoint to a storage operation and a subsequent detection
- Apply PRSA-based synthesis to modified sequencing graph



Incorporation of a checkpoint in a sequencing graph (*Det.* refers to detection operation)

95

Software for Rollback Recovery

- Map bioassay synthesis results to software in micro-controller memory
- A re-execution subroutine corresponds to a fragment of program (subprogram)

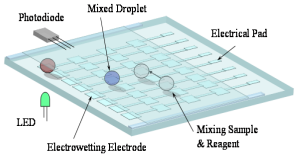
Subprogram for checkpoint C_2

Address	Fluidic operation	Duration (seconds)	Resource	Module placement
0083	O_0	0-6	4-electrode mixer	(2,2)
0084	C_1	7-12	Detector 1	(1,1)
0085	O_1	13-21	2x3-array dilutor	(3,3)
0086	O_2	22-27	2x4-array dilutor	(2,4)
0087	C_2	28-33	Detector 1	(1,1)
0088	O_5	7-15	2x3-array dilutor	(5,6)
0089	C_3	16-21	Detector 2	(10,1)
0090	O_3	30-35	2x4-array dilutor	(6,2)
0091	O_4	36-42	4-electrode mixer	(4,6)

Software corresponding to the bioassay synthesis result

96

Implementation for Rollback Recovery at Checkpoint C_2



Microfluidic Array

Bioassay Instructions

Bioassay Results

Micro-controller
(software programs)

Time: clock cycle 28
Instruction: start 0087 (C_2)

Time: clock cycle 33
Result: error detected at C_2

Time: clock cycle 33
Instruction:
(1) stop 0090 (O_3)
(2) stop time counter
(3) start 0085(O_1) to 0087(C_2)

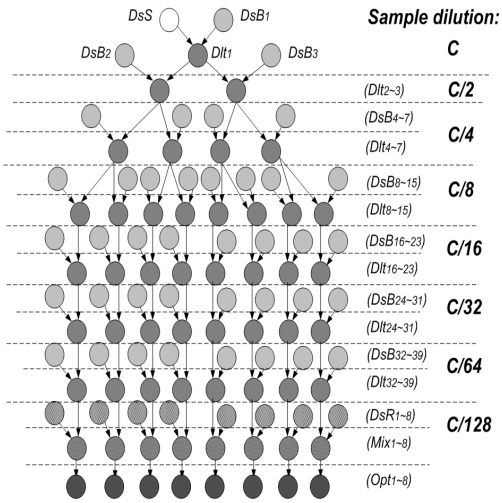
Time: clock cycle 33
Result: no error at C_2

Time: clock cycle 33
Instruction:
(1) resume following bioassay
(2) resume time counter

97

Evaluation Results – Protein Assay

A large-scale protein assay



V. Srinivasan et al., *Proc. SPIE* 2004

- 103 fluidic operations
  - Buffer dispensing: $DsB_{1\text{-}39}$
  - Reagent dispensing: $DsR_{1\text{-}8}$
  - Binary dilution: $Dlt_{1\text{-}39}$
  - Mixing operation: $Mix_{1\text{-}8}$
  - Optical detection: $Opt_{1\text{-}8}$
- Error-limit threshold
  - $E_{threshold}$ = 15%
- Checkpoint assignment
  - From sample dilution *C* to sample dilution *C/64*

98

98

Control Software for Protein Assay

- Map control-path-based protein assay synthesis results to software program in micro-controller memory
- C_4 to C_7 are checkpoints for operations Dlt_4 to Dlt_7

Subprogram for checkpoint C_5

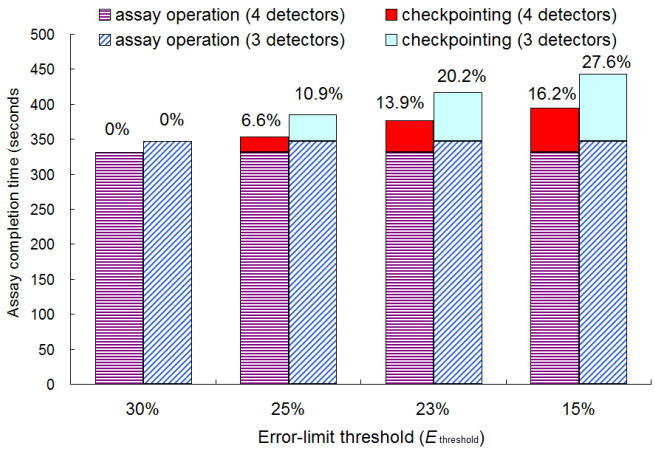
Address	Fluidic operation	Duration (seconds)	Resource	Module placement
0011	Dlt_4	46-53	4-electrode dilutor	(3,1)
0012	C_4	54-59	Detector 1	(1,1)
0013	Dlt_5	76-81	2x4-array dilutor	(5,3)
0014	C_5	82-87	Detector 3	(5,1)
0015	Dlt_6	56-61	2x4-array dilutor	(1,5)
0016	C_6	62-67	Detector 1	(1,1)
0017	Dlt_7	58-70	2x2-array dilutor	(5,3)
0018	C_7	71-76	Detector 2	(1,10)

Software corresponding to the bioassay synthesis

99

Assay Completion Time (No Error)

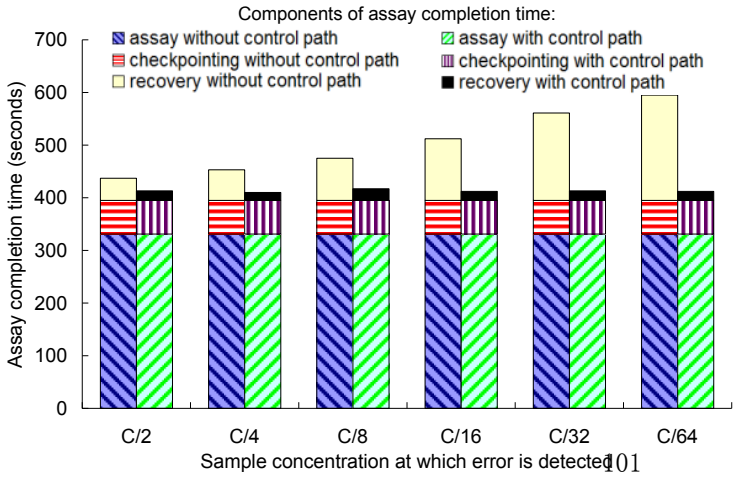
- Completion time for control-path-based protein assay when no error occurs during bioassay
  - 4-detector case and 3-detector case
  - Completion time = assay time + checkpointing time



100

Assay Completion Time (With Error)

- Completion time for the scheduled protein assay protocol with and without control paths
  - Errors appear at intermediate points (sample concentrations)
  - Completion time = assay time + checkpointing time + recovery time

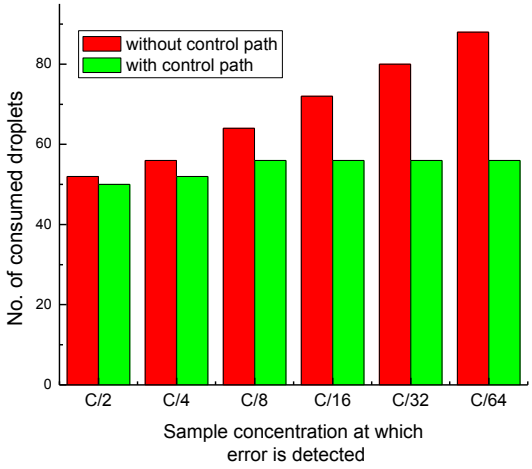


101

101

Droplet Consumption (With Error)

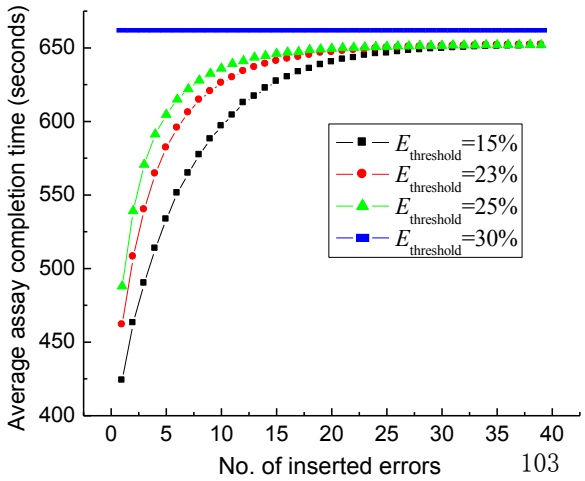
- No. of droplets consumed for the scheduled protein assay protocol with and without control paths
  - Errors appear at intermediate points (sample concentrations)



102

Assay Completion Time for Randomly Injected Errors

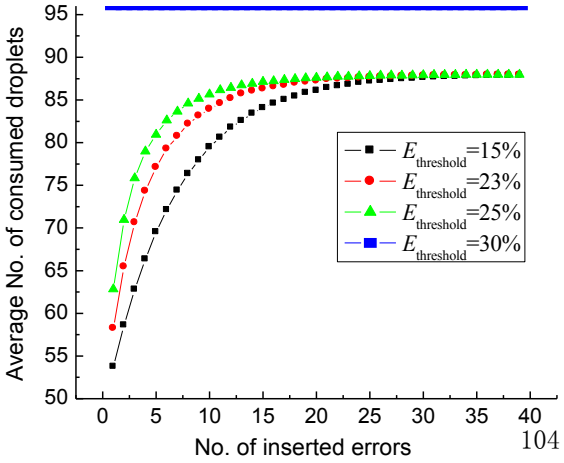
- Average assay completion time for the scheduled protein assay protocol
  - Different numbers of errors are injected at randomly chosen dilutions
  - Results for various error-limit thresholds: 15%, 23%, 25%, 30%



103

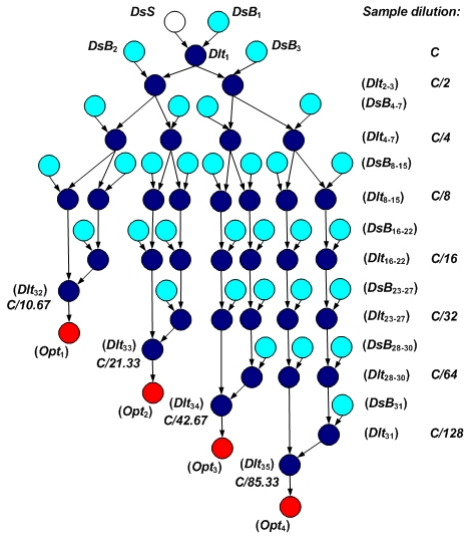
Droplet Consumption for Randomly Injected Errors

- Average number of consumed droplets for the scheduled protein assay protocol
  - Different numbers of errors are injected at randomly chosen dilutions
  - Results for various error-limit thresholds: 15%, 23%, 25%, 30%



104

Evaluation Result – Interpolating Mixing Architecture



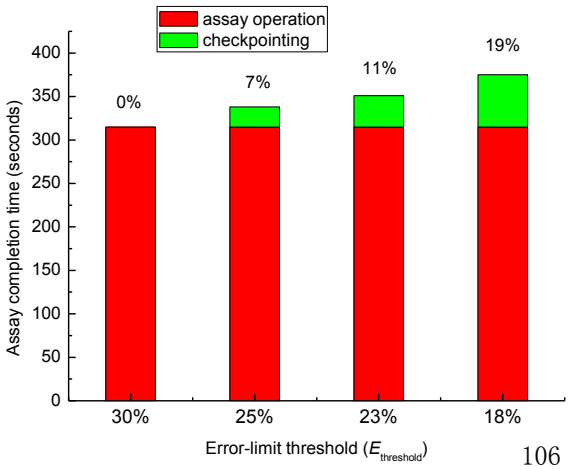
- 71 fluidic operations
  - Buffer dispensing: $DsB_{1\text{-}31}$
  - Dilution operation: $Dlt_{1\text{-}35}$
  - Optical detection: $Opt_{1\text{-}4}$
- Error-limit threshold
  - $E_{threshold}$ = 18%
- Checkpoint assignment
  - From sample dilutions $C/2^N$, C/10.67, C/21.33, C/42.67, and C/85.33

H. Ren et al., *Transducers* 2003

105

Assay Completion Time (No Error)

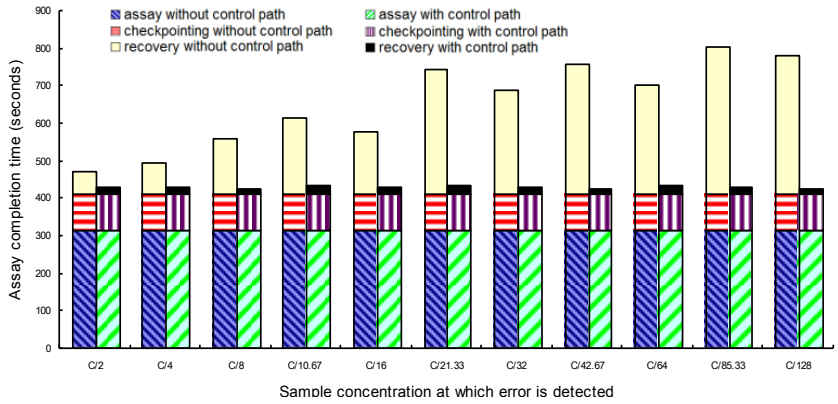
- Completion time for control-path-based interpolating mixing architecture when no error occurs during bioassay
  - Completion time = assay time + checkpointing time



106

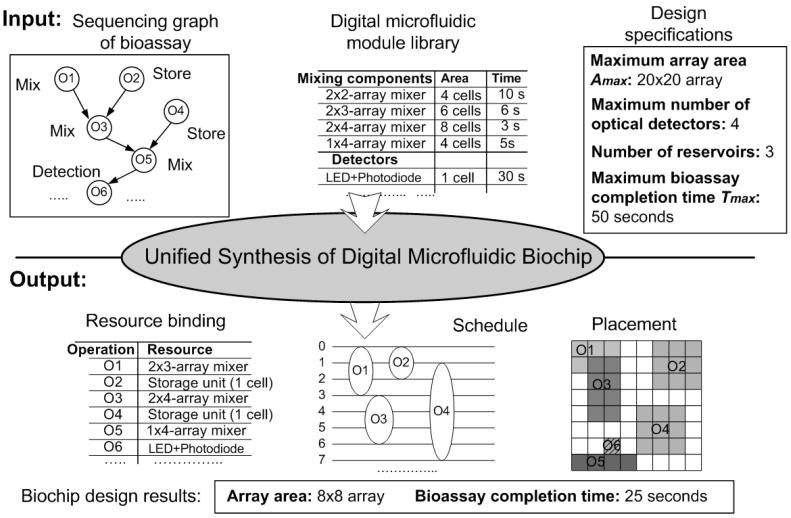
Assay Completion Time (With Error)

- Completion time for the scheduled interpolating mixing architecture with and without control paths
  - Errors appear at intermediate points (sample concentrations)
  - Completion time = assay time + checkpointing time + recovery time

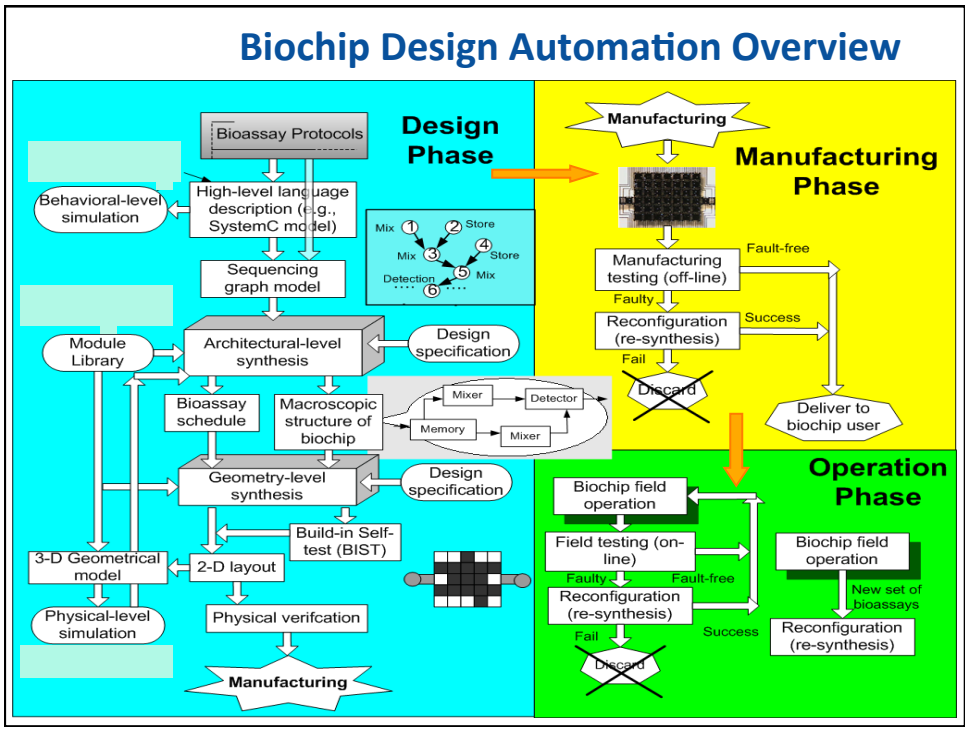


107

System-Level Design of Microfluidic Biochips



108


Biochip Design Automation Overview
Design Phase
Manufacturing Phase
Operation Phase
Bioassay Protocols
Behavioral-level simulation
High-level language description (e.g., SystemC model)
Sequencing graph model
Module Library
Architectural-level synthesis
Design specification
Bioassay schedule
Macroscopic structure of biochip
Mixer
Detector
Memory
Mixer
Geometry-level synthesis
Design specification
Build-in Self-test (BIST)
3-D Geometrical model
2-D layout
Physical-level simulation
Physical verifcation
Manufacturing
Mix
Store
Mix
Store
Detection
Mix
Manufacturing
Manufacturing testing (off-line)
Fault-free
Faulty
Reconfiguration (re-synthesis)
Success
Fail
Discard
Deliver to biochip user
Biochip field operation
Field testing (on-line)
Faulty
Fault-free
Reconfiguration (re-synthesis)
Fail
Discard
Success
Biochip field operation
New set of bioassays
Reconfiguration (re-synthesis)