Computer-Aided Design for Microfluidic Chips Based on Multilayer Soft Lithography

Nada Amin[1], William Thies[2], Saman Amarasinghe[1]

[1] Massachusetts Institute of Technology

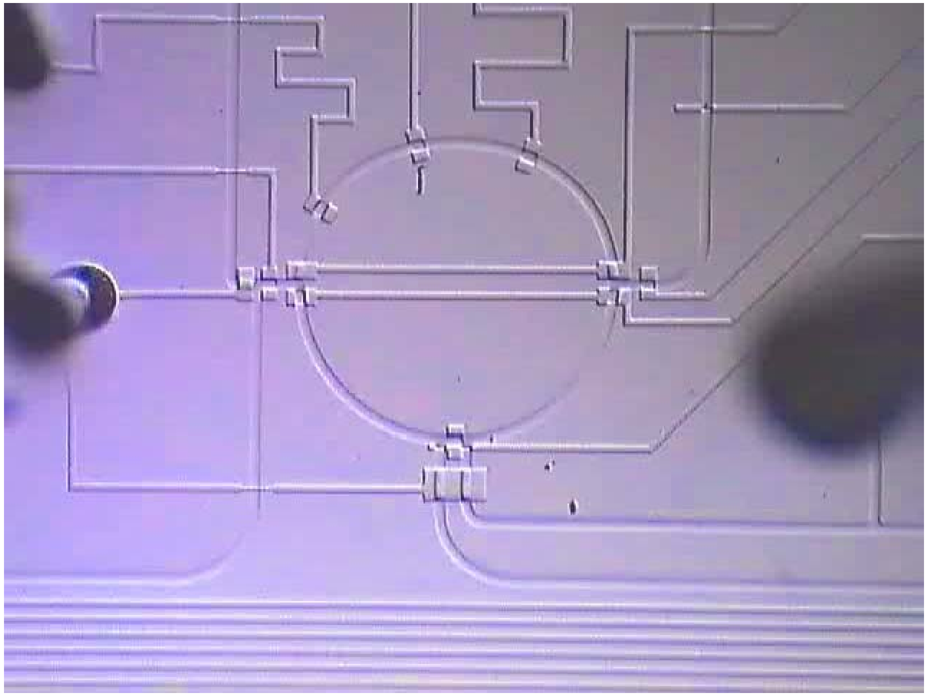
[2] Microsoft Research India

International Conference on Computer Design

October 5, 2009

Microfluidic Chips

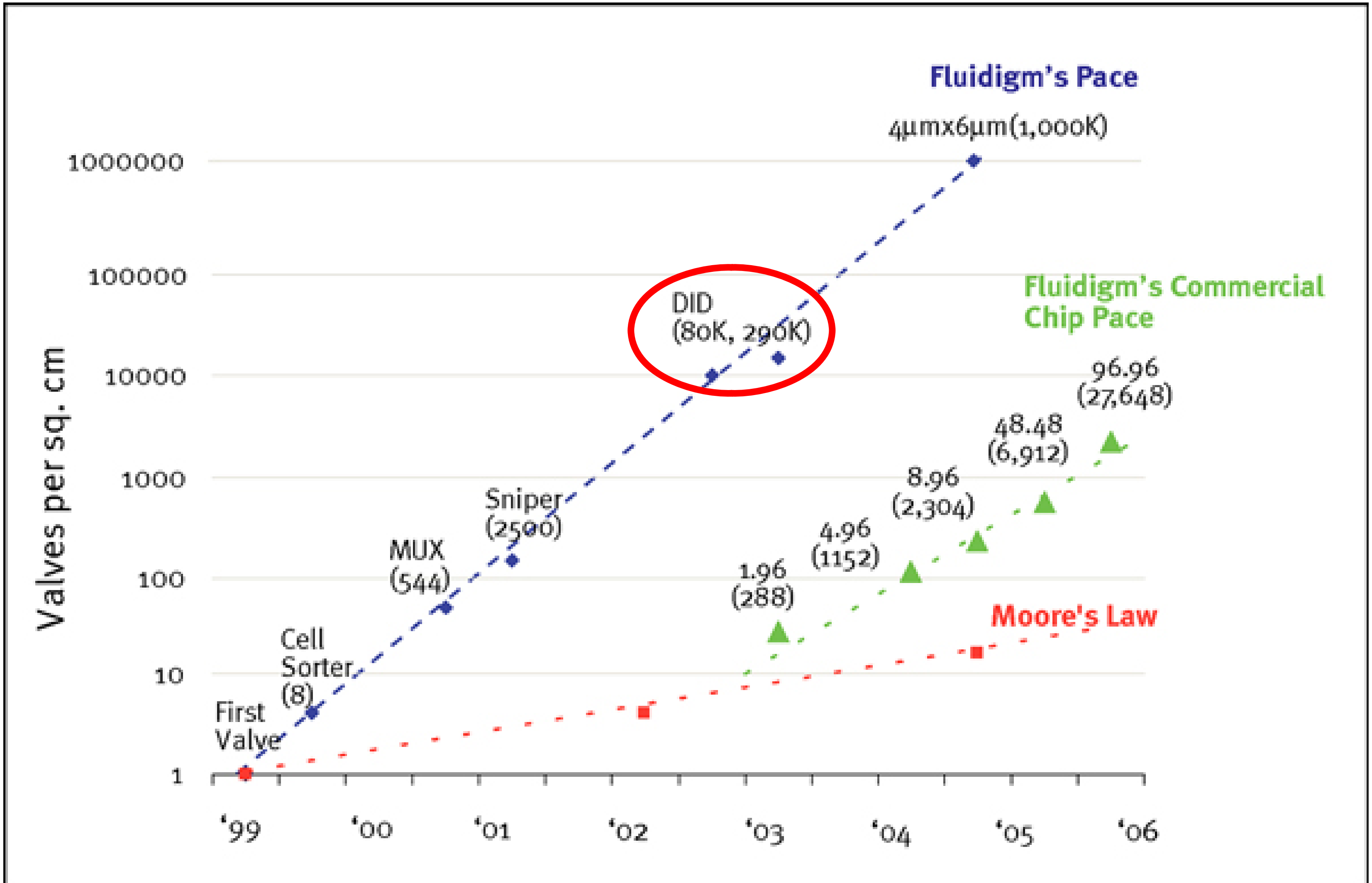
- **Idea: a whole biology lab on a single chip**
  - Input/output
  - **Sensors**: pH, glucose, temperature, etc.
  - **Actuators**: mixing, PCR, electrophoresis, cell lysis, etc.



1 mm *10x real-time*

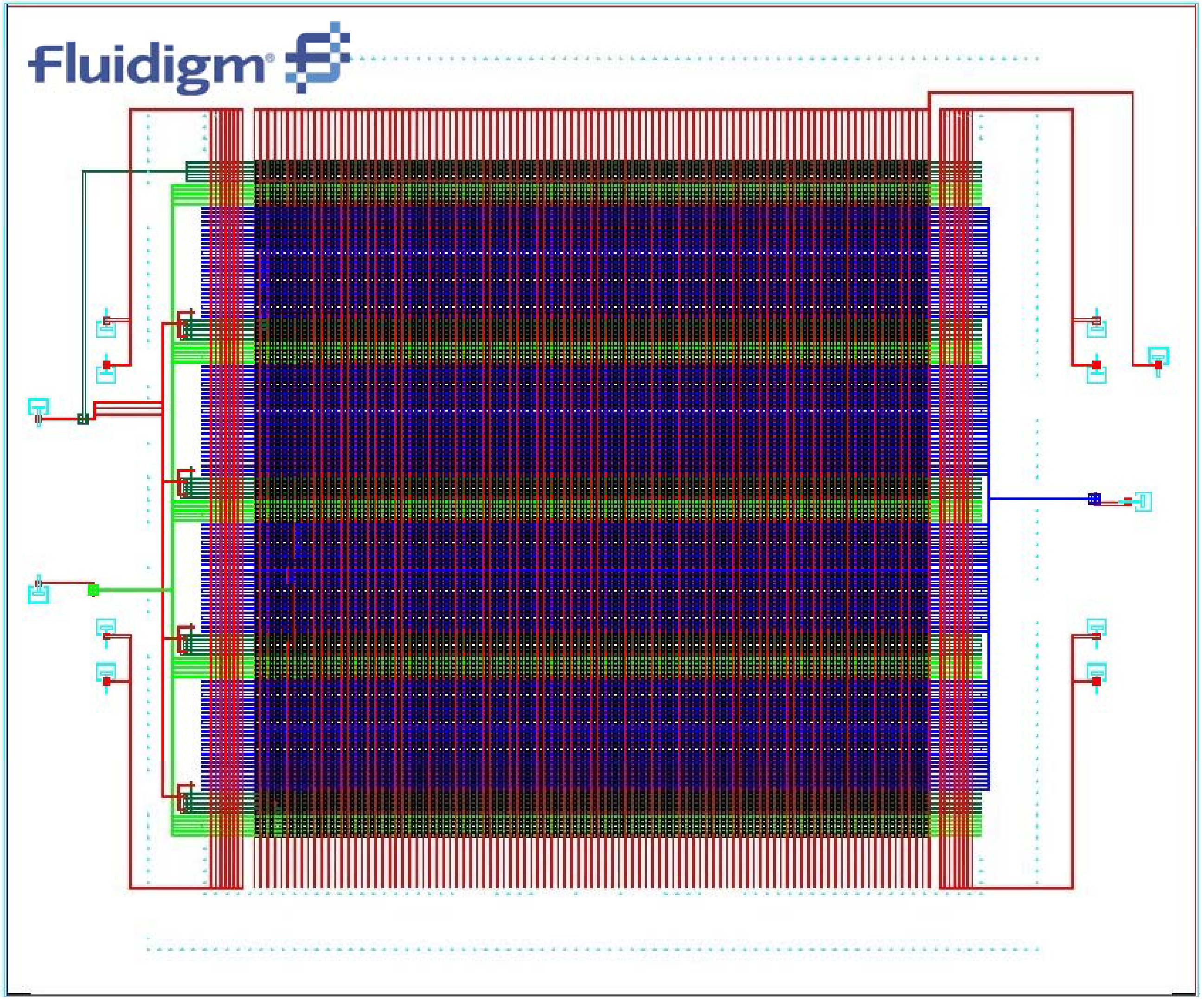
- **Benefits:**
  - Small sample volumes
  - High throughput
  - Low-cost
- **Applications:**
  - Biochemistry - Cell biology
  - Biological computing

Moore's Law of Microfluidics: Valve Density Doubles Every 4 Months



Source: Fluidigm Corporation (http://www.fluidigm.com/images/mlaw_lg.jpg)

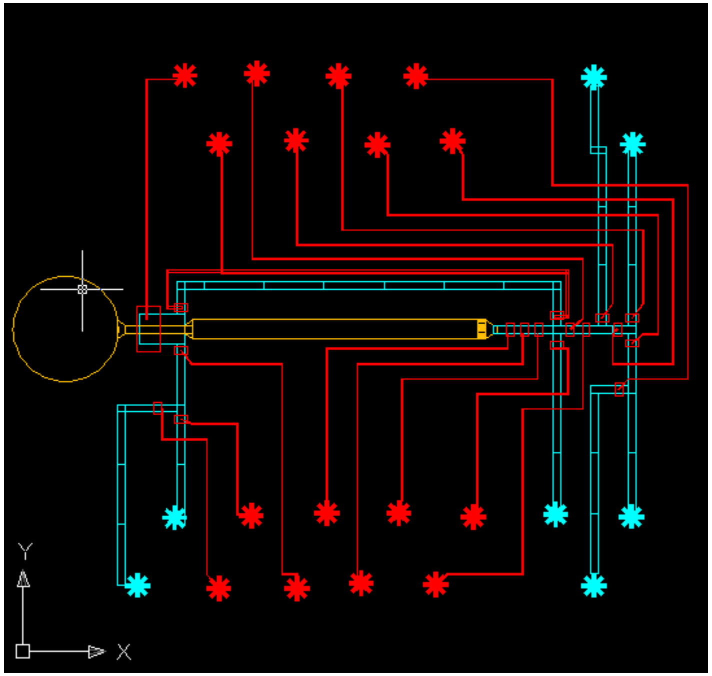
Moore’s Law of Microfluidics: Valve Density Doubles Every 4 Months



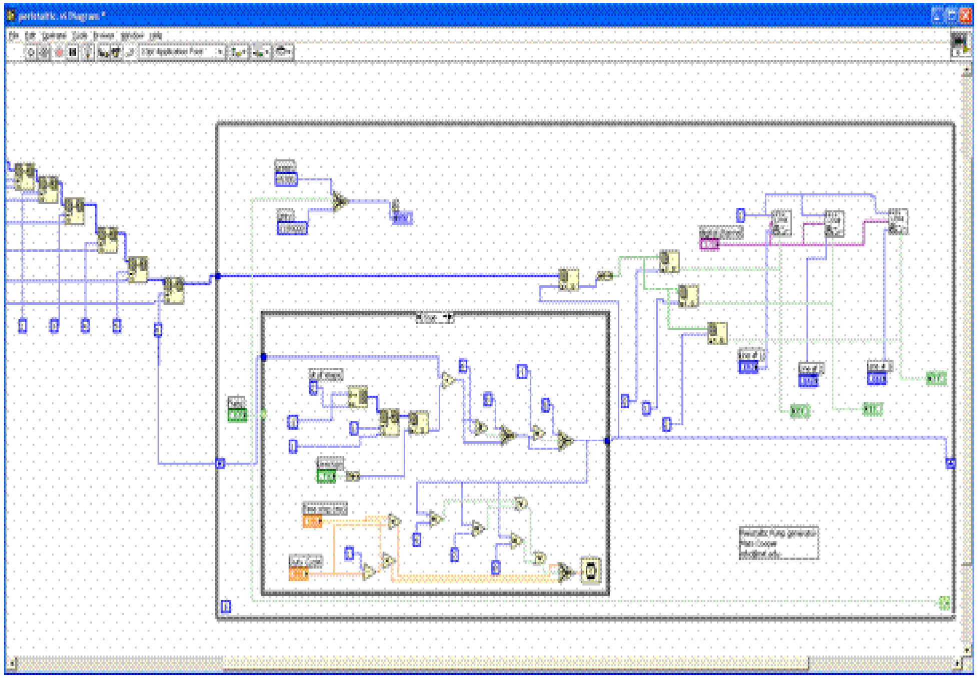
Source: Fluidigm Corporation (http://www.fluidigm.com/didIFC.htm)

Current Practice: Manage Gate-Level Details from Design to Operation

- For every change in the experiment or the chip design:



fabricate chip



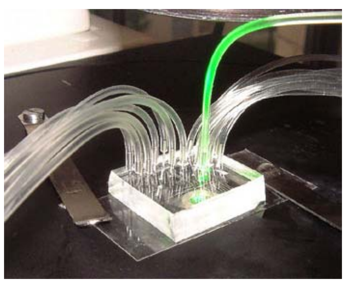
1. Manually draw in AutoCAD

2. Operate each gate from LabView

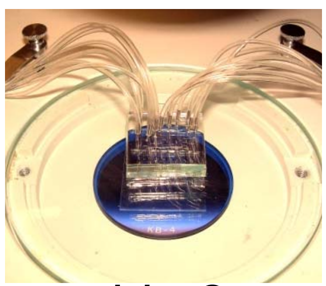
Abstraction Layers for Microfluidics

Protocol Description Language
- architecture-independent protocol description

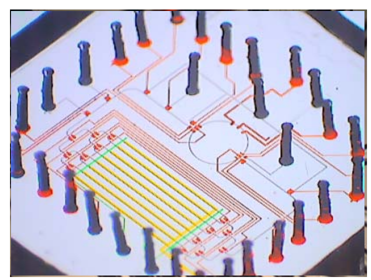
Fluidic Instruction Set Architecture (ISA)
- primitives for I/O, storage, transport, mixing



chip 1



chip 2



chip 3

Fluidic Hardware Primitives
- valves, multiplexers, mixers, latches

Silicon Analog

C

x86

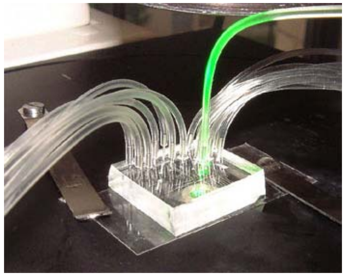
Pentium III, Pentium IV

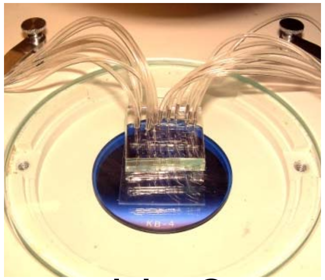
transistors, registers, …

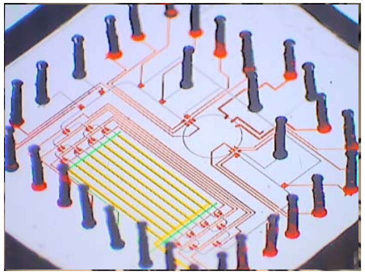
Abstraction Layers for Microfluidics

Protocol Description Language
- architecture-independent protocol description

Fluidic Instruction Set Architecture (ISA)
- primitives for I/O, storage, transport, mixing


chip 1


chip 2


chip 3

Fluidic Hardware Primitives
- valves, multiplexers, mixers, latches

Contributions

BioStream Language
[IWBDA 2009]

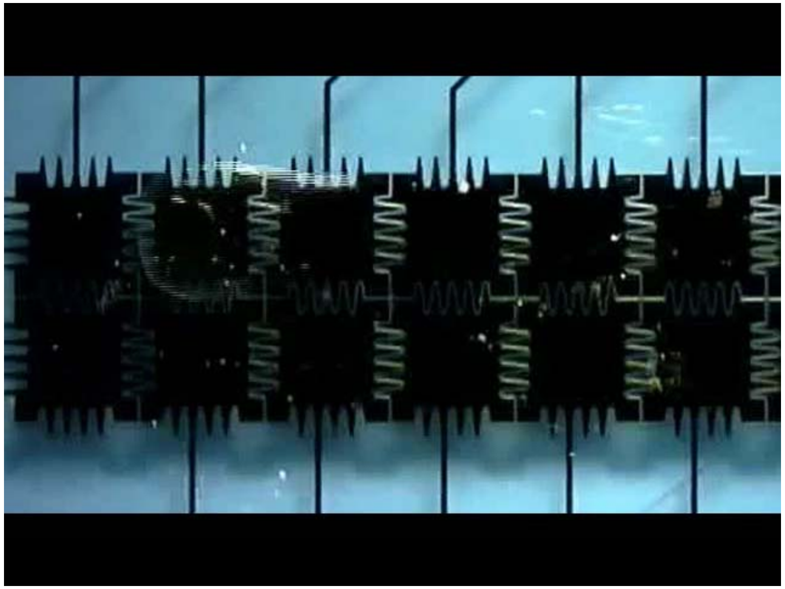
Optimized Compilation
[Natural Computing 2007]

Demonstrate Portability
[DNA 2006]

Micado AutoCAD Plugin
[MIT 2008, ICCD 2009]

Digital Sample Control
Using Soft Lithography
[Lab on a Chip '06]

Droplets vs. Continuous Flow



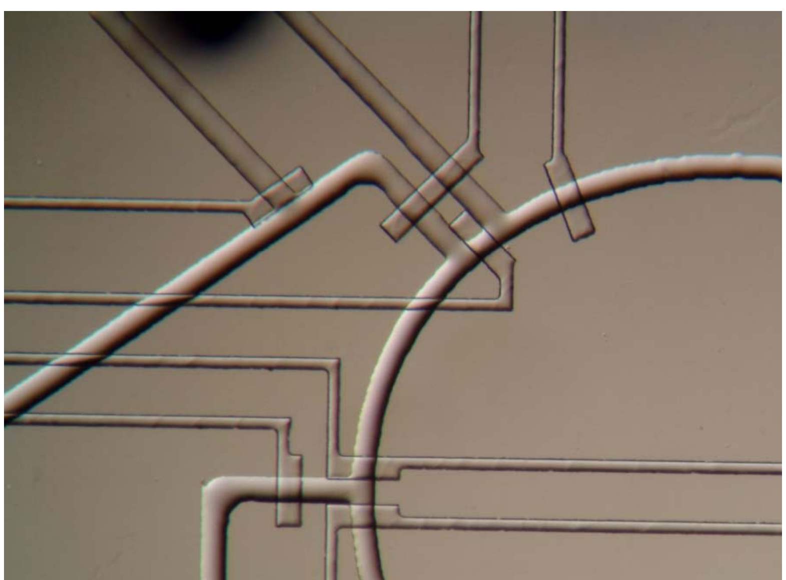
Source: Chakrabarty et al, Duke University

Digital manipulation of droplets on an electrode array

[Chakrabarty, Fair, Gascoyne, Kim, …]

Pro:

- Reconfigurable routing
- Electrical control
- More traction in CAD community



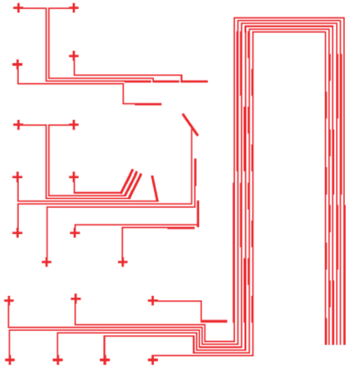
Continuous flow of fluids (or droplets) through fixed channels

[Whitesides, Quake, Thorsen, …]

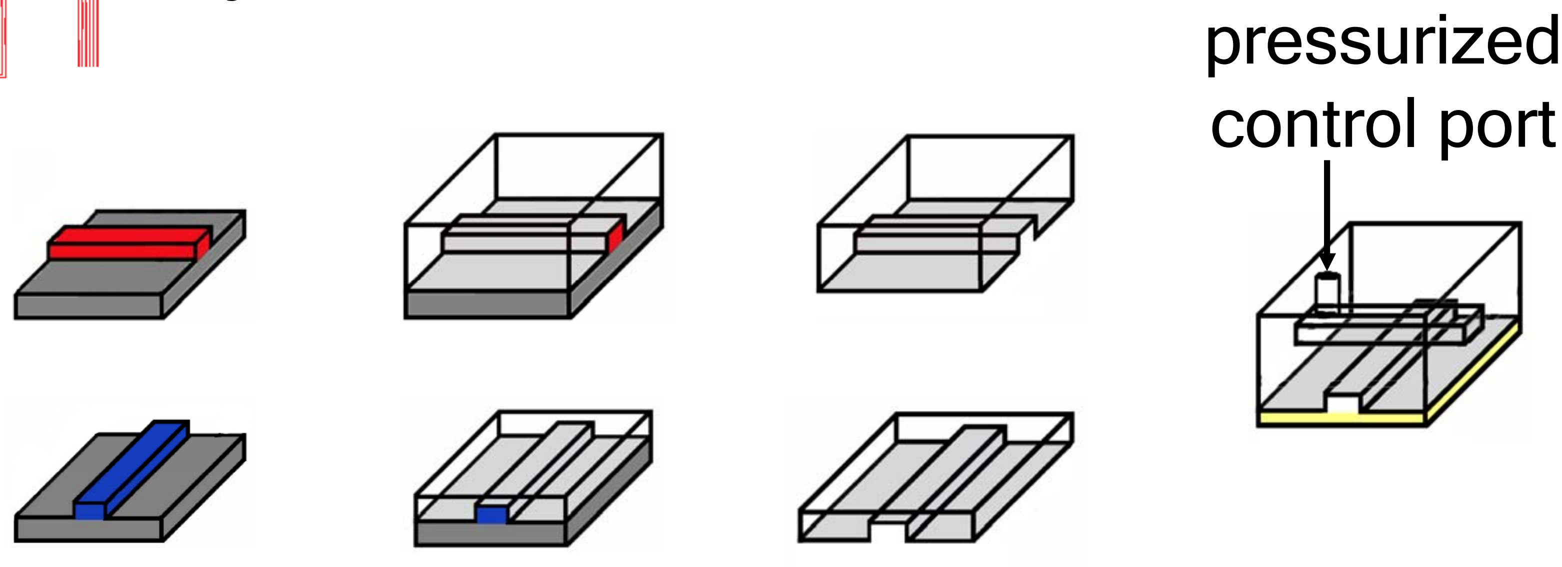
Pro:

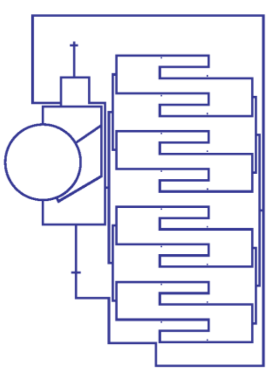
- Smaller sample sizes
- Made-to-order availability [Stanford]
- More traction in biology community

Primitive 1: A Valve (Quake et al.)



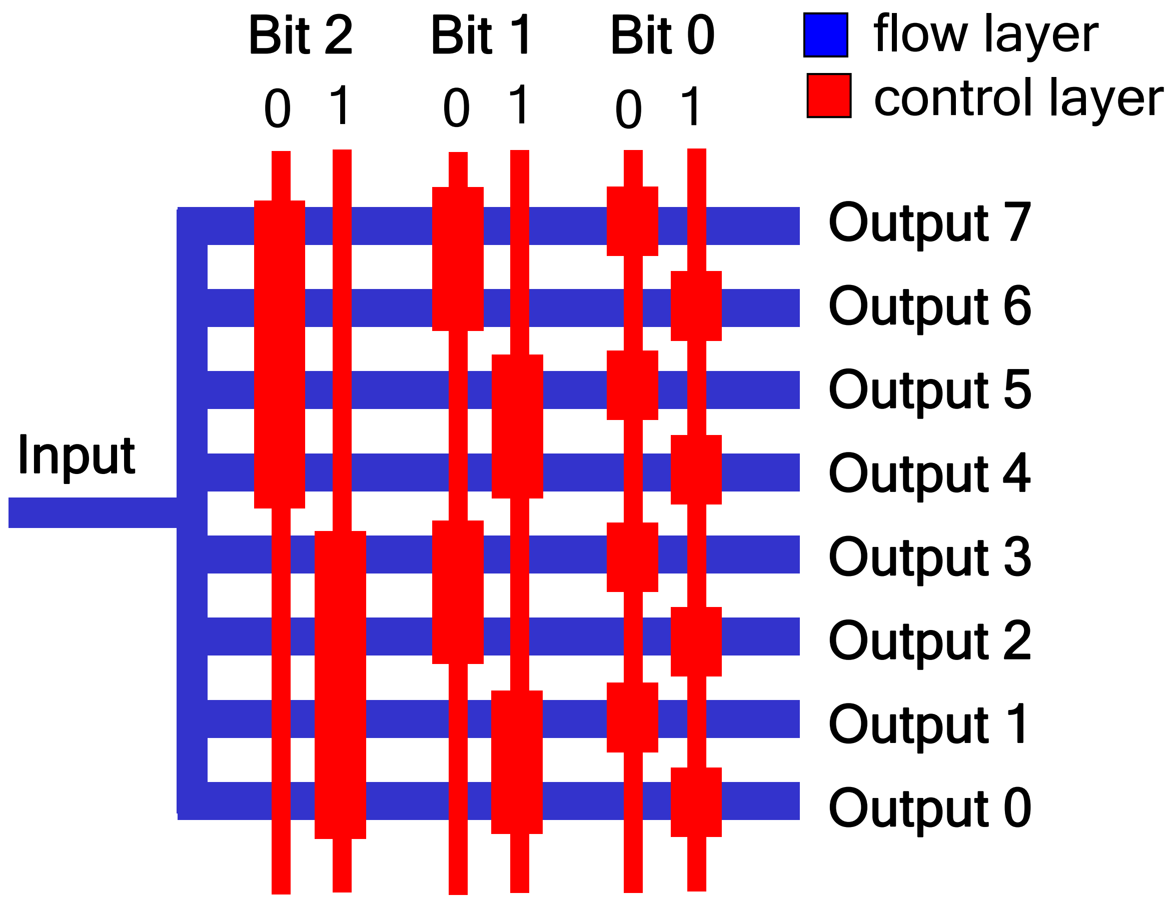
Control Layer





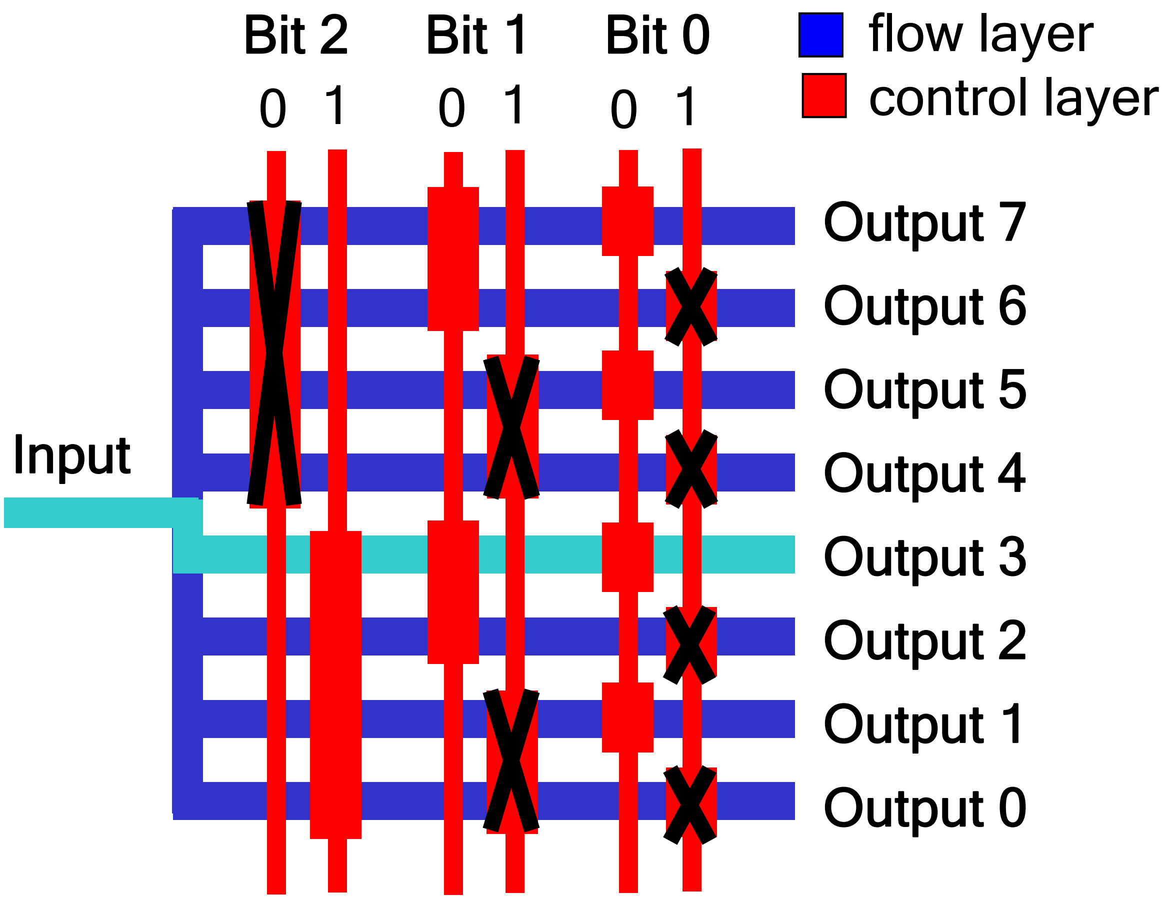
Flow Layer

Primitive 2: A Multiplexer (Thorsen et al.)



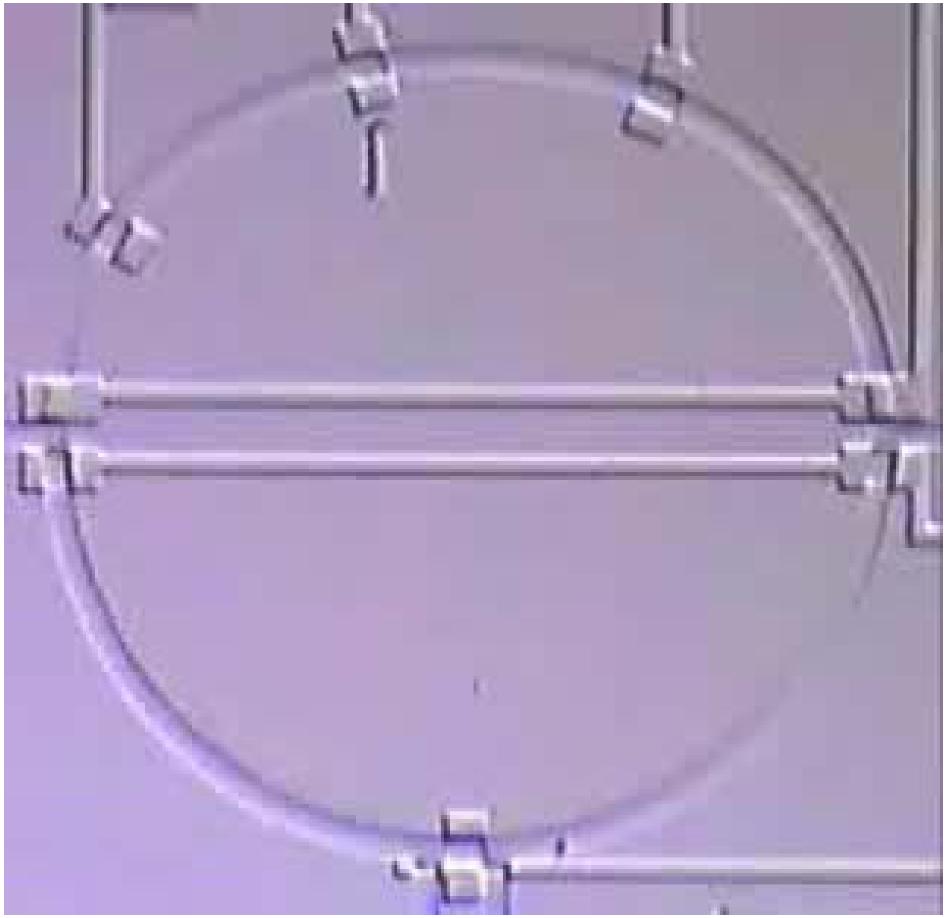
Example: select 3 = 011

Primitive 2: A Multiplexer (Thorsen et al.)



Example: select 3 = 011

Primitive 3: A Mixer (Quake et al.)



1. Load sample on bottom
2. Load sample on top
3. Peristaltic pumping

Rotary Mixing

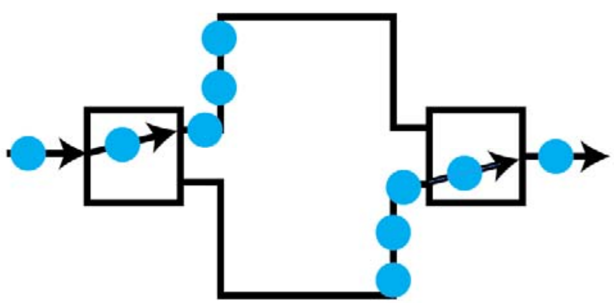
CAD Tools for Microfluidic Chips

- **Goal: automate placement, routing, control of microfluidic features**
- **Why is this different than electronic CAD?**

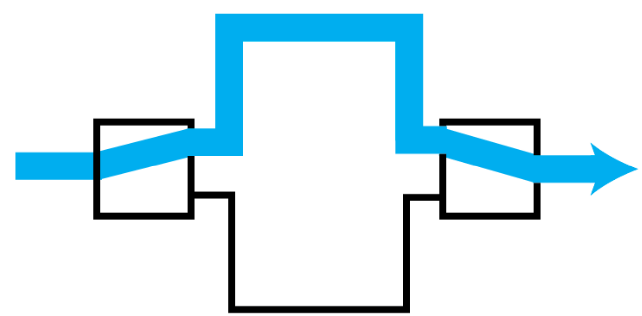
1. Control ports (I/O pins) are bottleneck to scalability

- – Pressurized control signals cannot yet be generated on-chip
- – Thus, each logical set of valves requires its own I/O port

2. Control signals correlated due to continuous flows



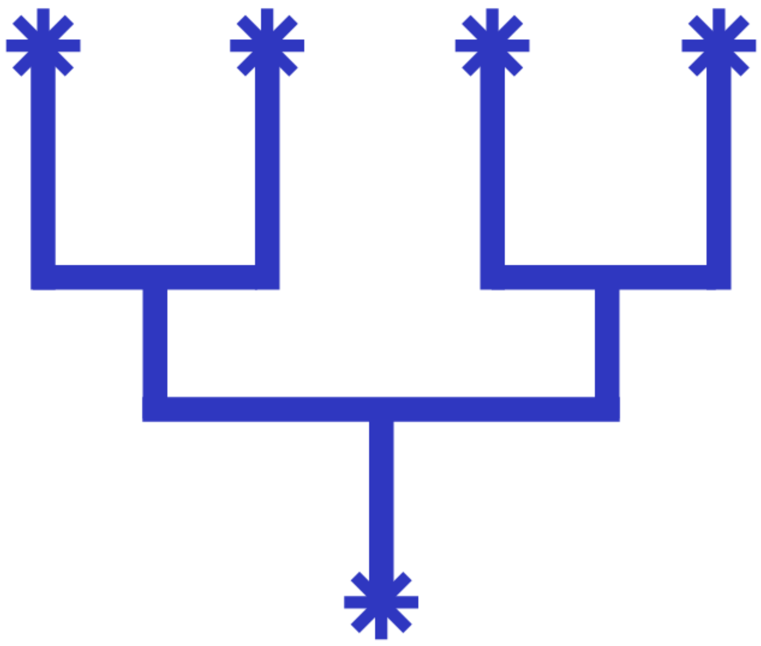
pipelined flow



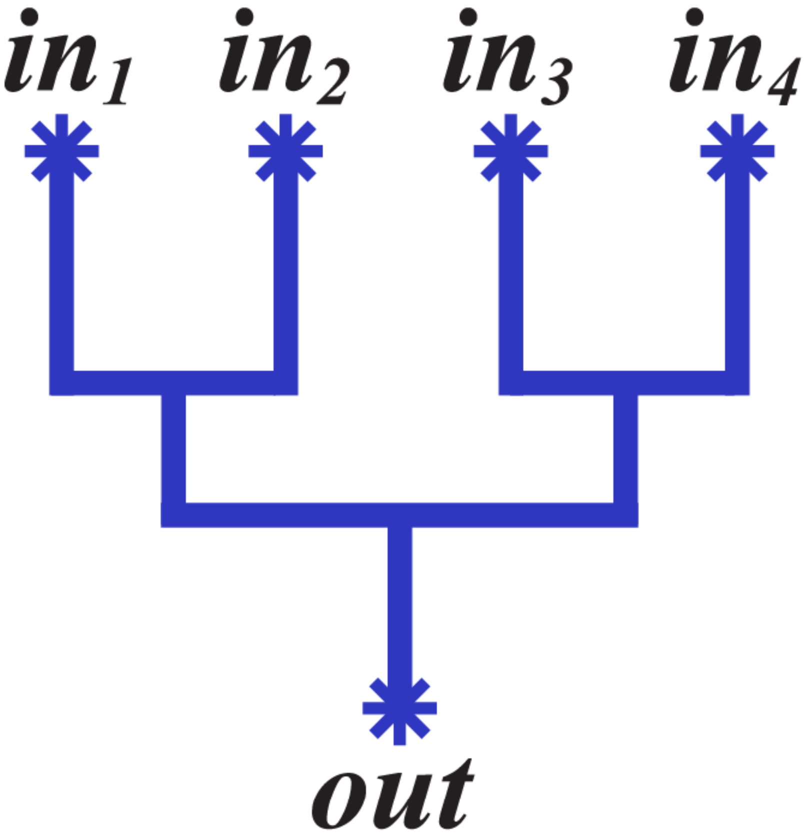
continuous flow

→ Demand & opportunity for minimizing control logic

Our Paper:
Automatic Generation of Control Layer



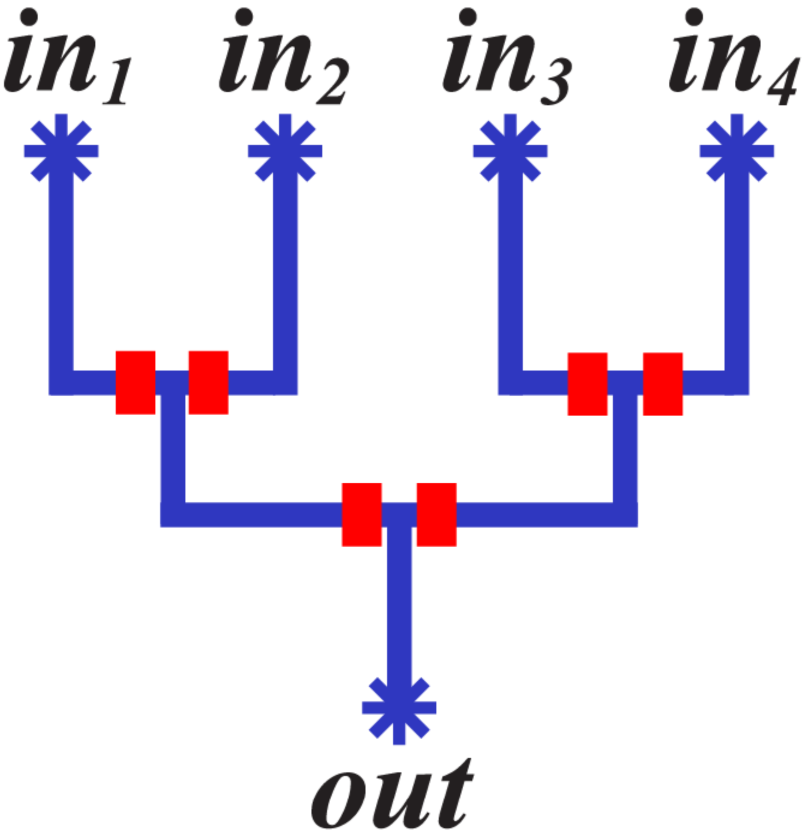
Our Paper: Automatic Generation of Control Layer



$$(in_1 \vee in_2 \vee in_3 \vee in_4) \rightarrow out$$

1. Describe Fluidic ISA

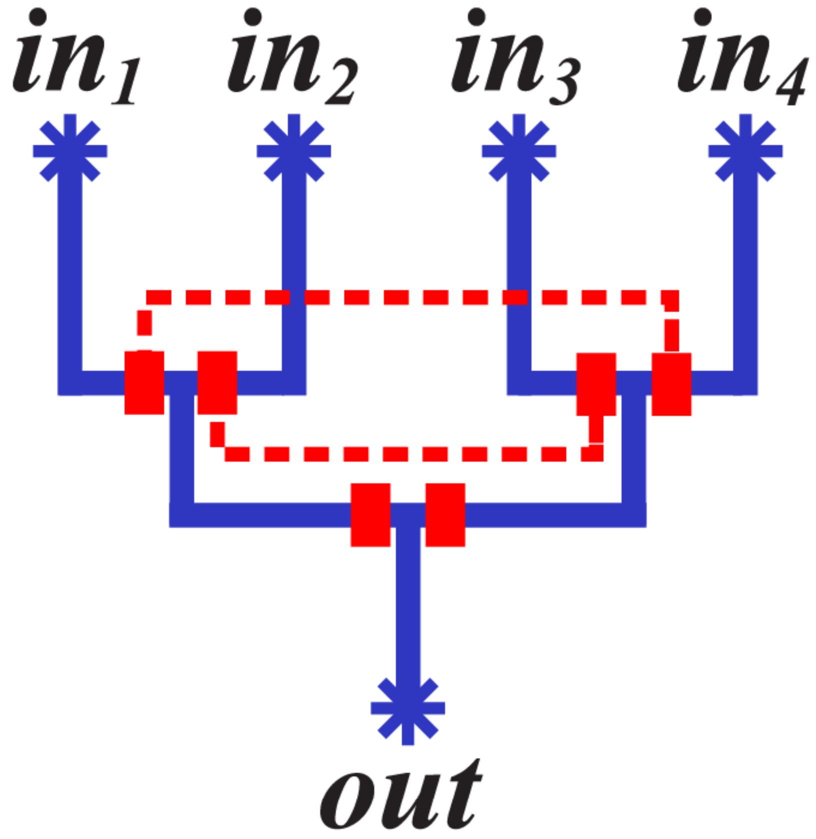
Our Paper:
Automatic Generation of Control Layer



$$(in_1 \vee in_2 \vee in_3 \vee in_4) \rightarrow out$$

1. Describe Fluidic ISA
2. Infer control valves

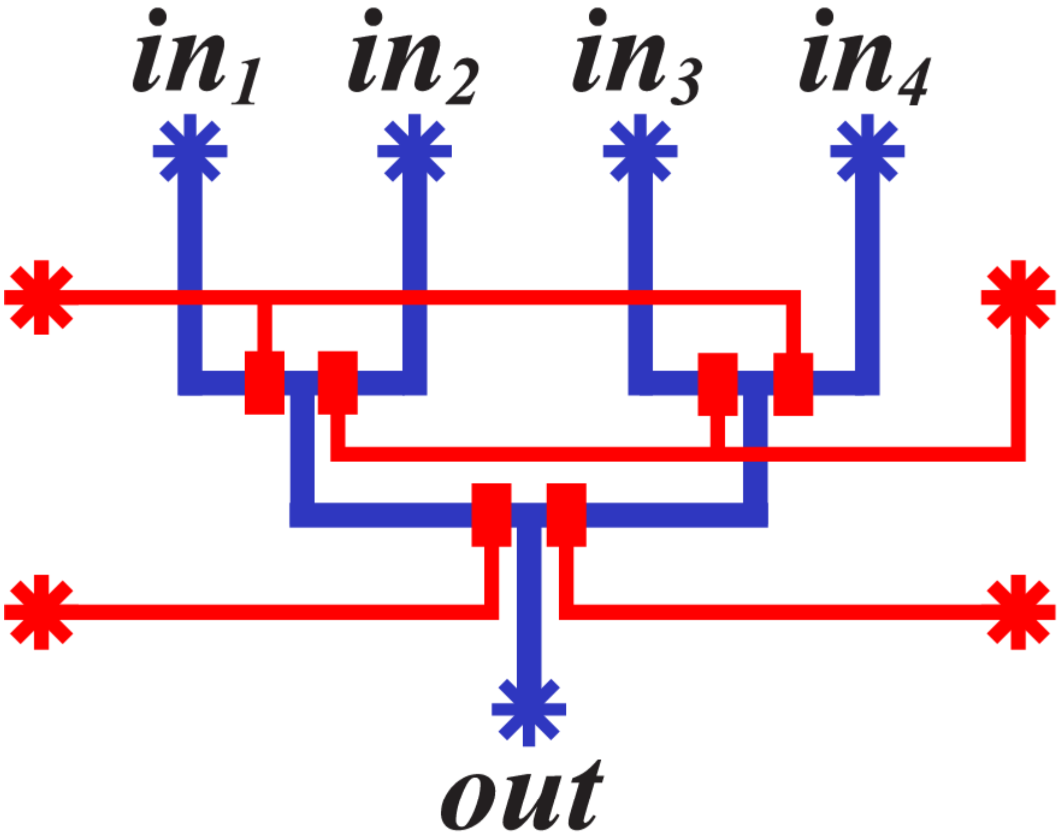
Our Paper:
Automatic Generation of Control Layer



1. Describe Fluidic ISA
2. Infer control valves
3. Infer control sharing

$$(in_1 \vee in_2 \vee in_3 \vee in_4) \rightarrow out$$

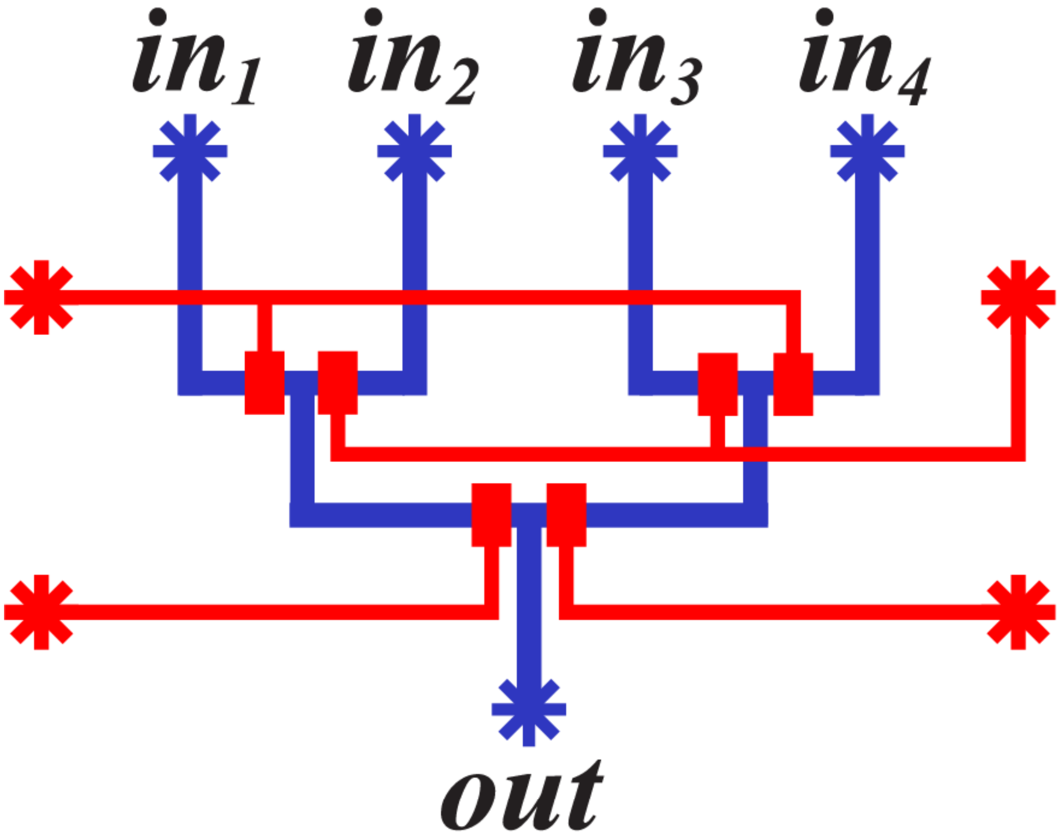
Our Paper:
Automatic Generation of Control Layer



1. Describe Fluidic ISA
2. Infer control valves
3. Infer control sharing
4. Route valves to control ports

$$(in_1 \vee in_2 \vee in_3 \vee in_4) \rightarrow out$$

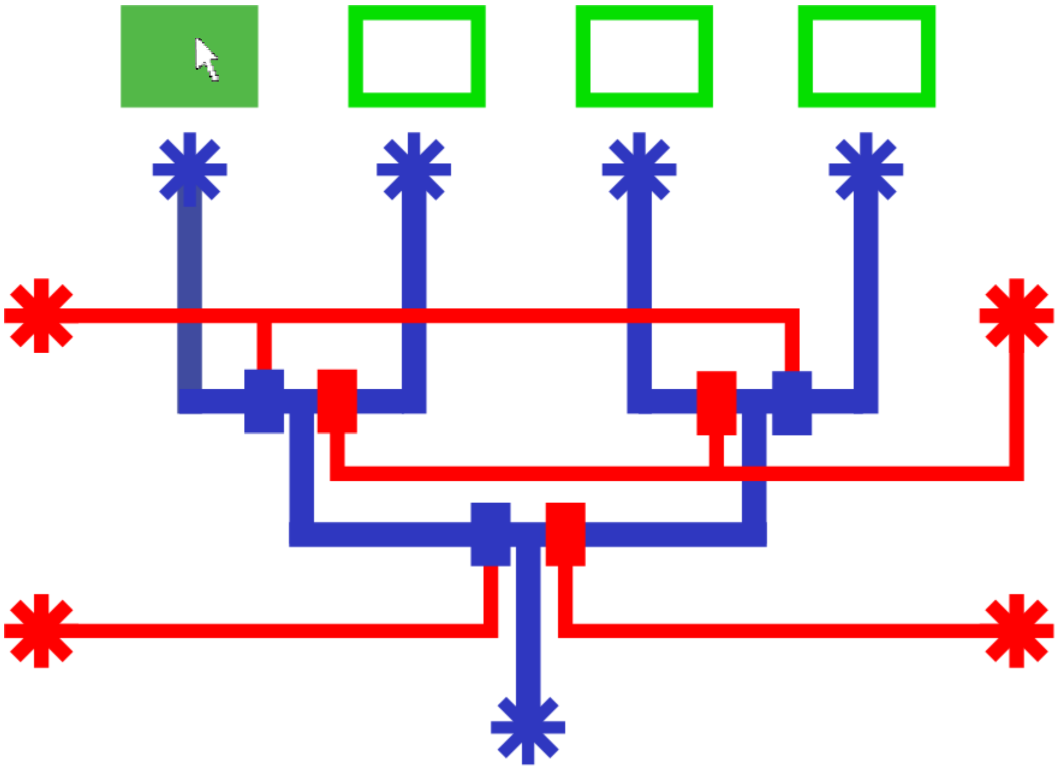
Our Paper:
Automatic Generation of Control Layer



1. Describe Fluidic ISA
2. Infer control valves
3. Infer control sharing
4. Route valves to control ports
5. Generate an interactive GUI

$$(in_1 \vee in_2 \vee in_3 \vee in_4) \rightarrow out$$

Our Paper:
Automatic Generation of Control Layer

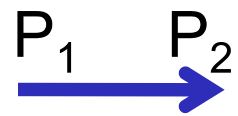


1. Describe Fluidic ISA
2. Infer control valves
3. Infer control sharing
4. Route valves to control ports
5. Generate an interactive GUI

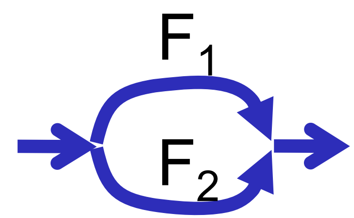
1. Describe a Fluidic ISA

- **Hierarchical and composable flow declarations**

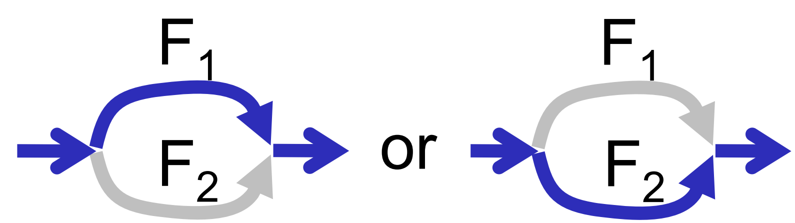
Sequential flow — $P_1 \rightarrow P_2$



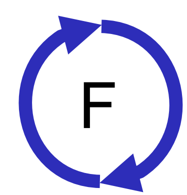
AND-flow — $F_1 \wedge F_2$



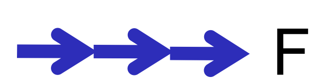
OR-flow — $F_1 \vee F_2$



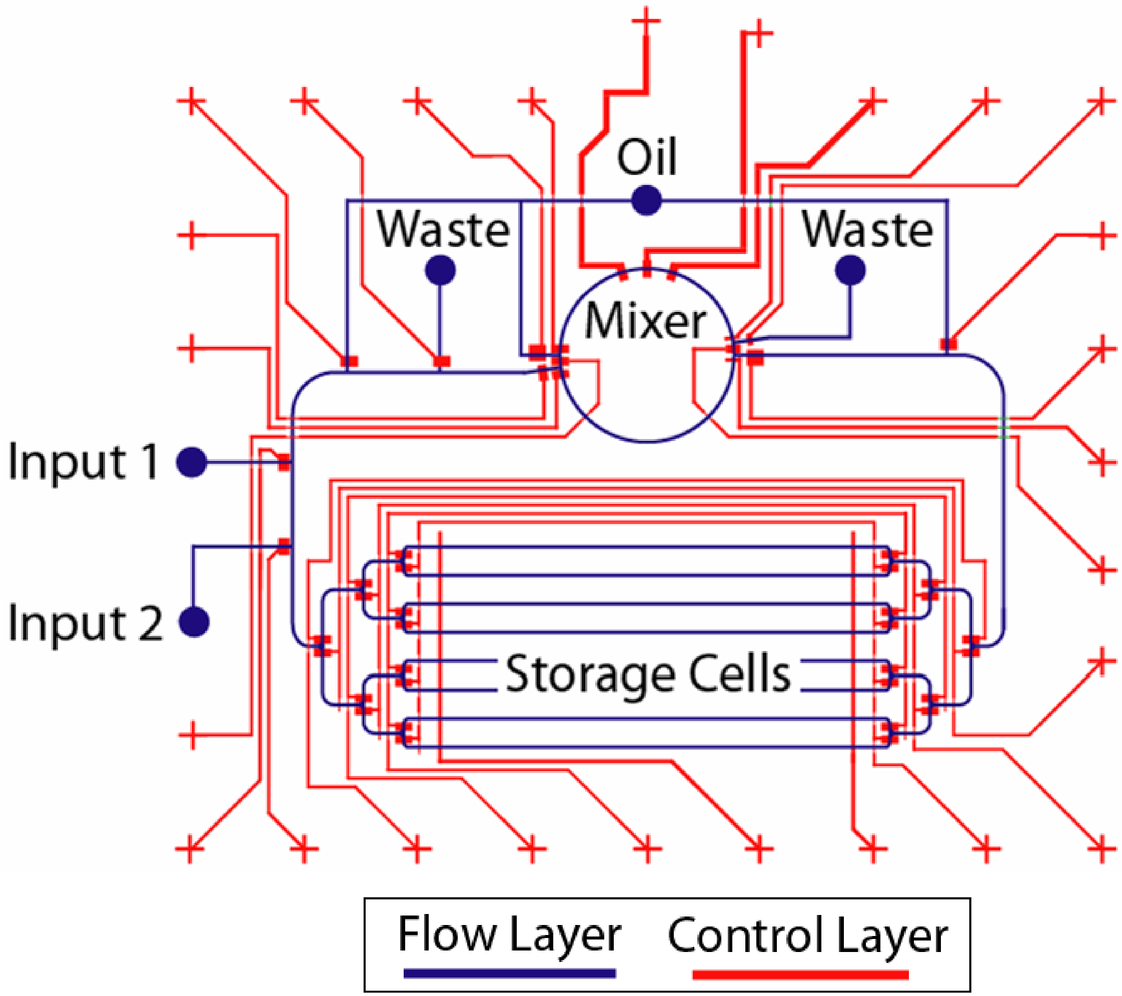
Mixing — mix(F)



Pumped flow — pump(F)



1. Describe a Fluidic ISA

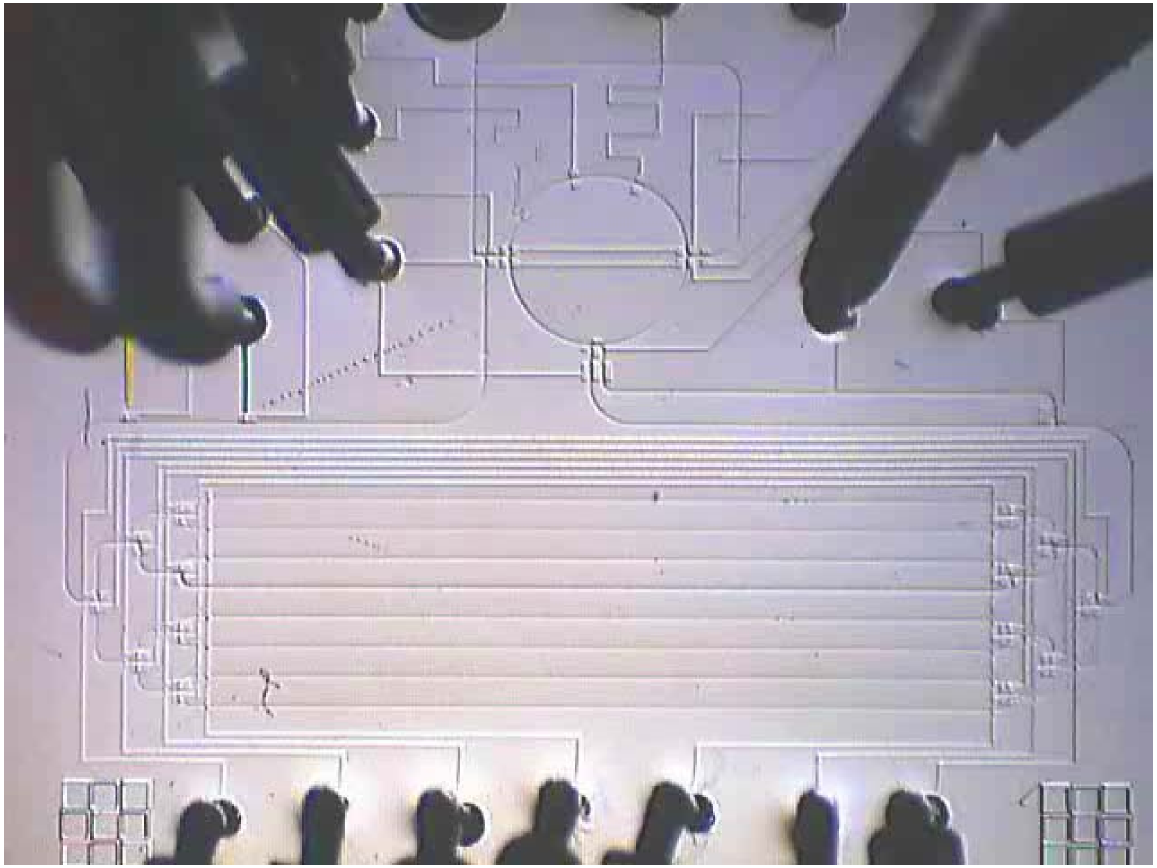


1. Describe a Fluidic ISA

mix-and-store (S_1, S_2, D) {

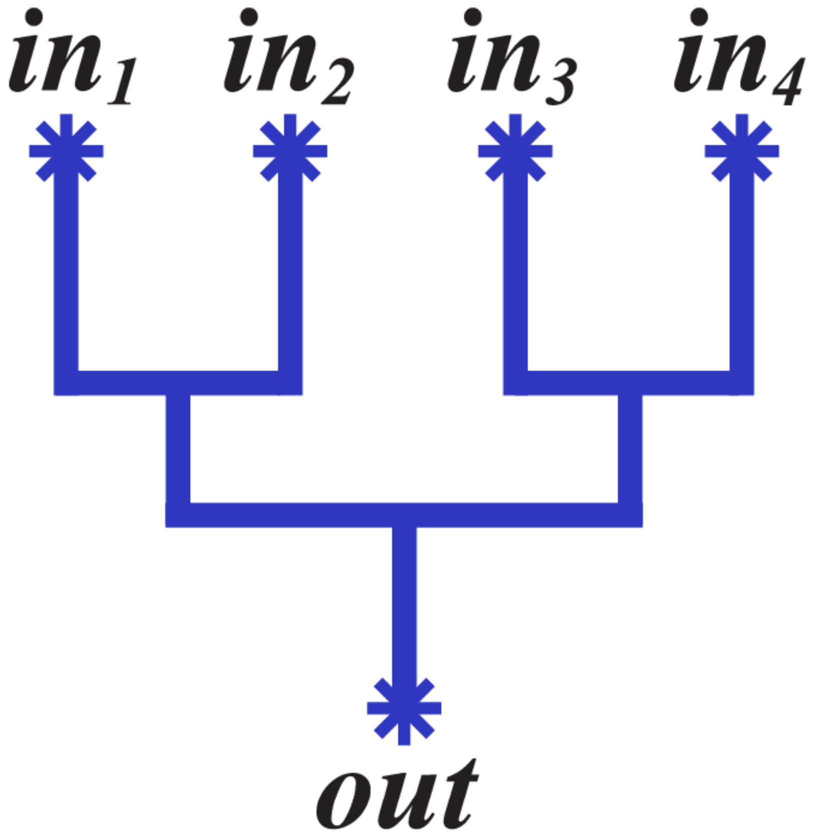
1. in_1 → top → out
2. in_2 → bot → out
3. **mix**(top → bot-left → bot-right → top)
4. wash → bot-right → top → bot-left → store

}

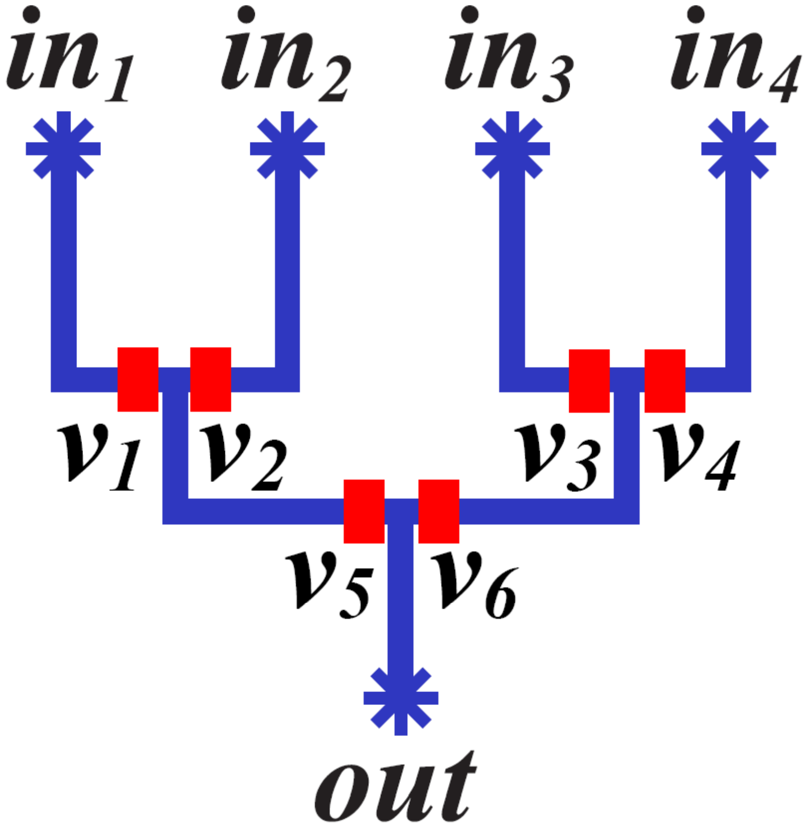


50x real-time

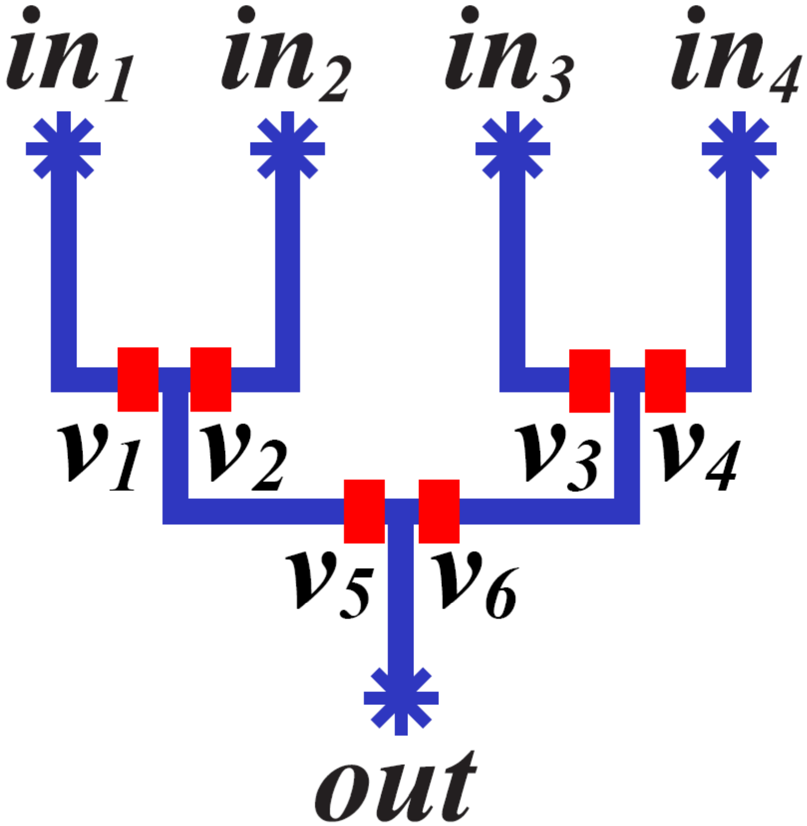
2. Infer Control Valves



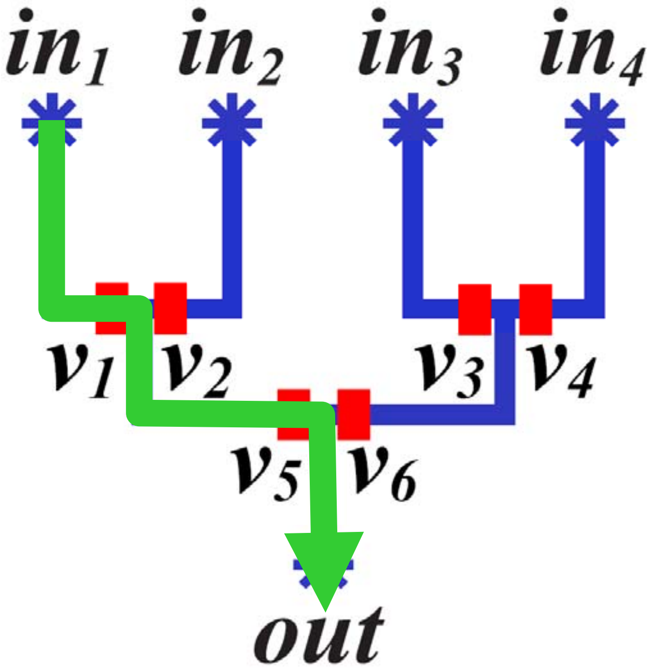
2. Infer Control Valves



3. Infer Control Sharing

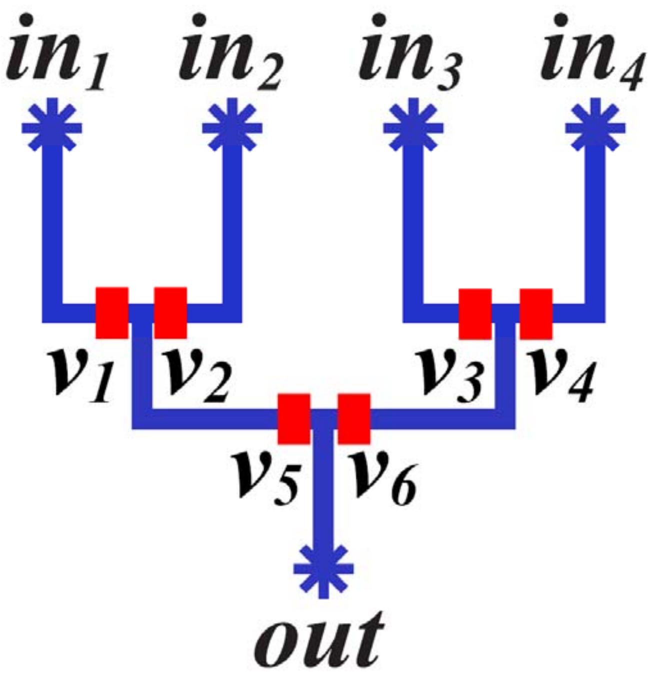


3. Infer Control Sharing



	$in_1 \rightarrow o$	$in_2 \rightarrow o$	$in_3 \rightarrow o$	$in_4 \rightarrow o$
v_1	open			
v_2	closed			
v_3				
v_4				
v_5	open			
v_6	closed			

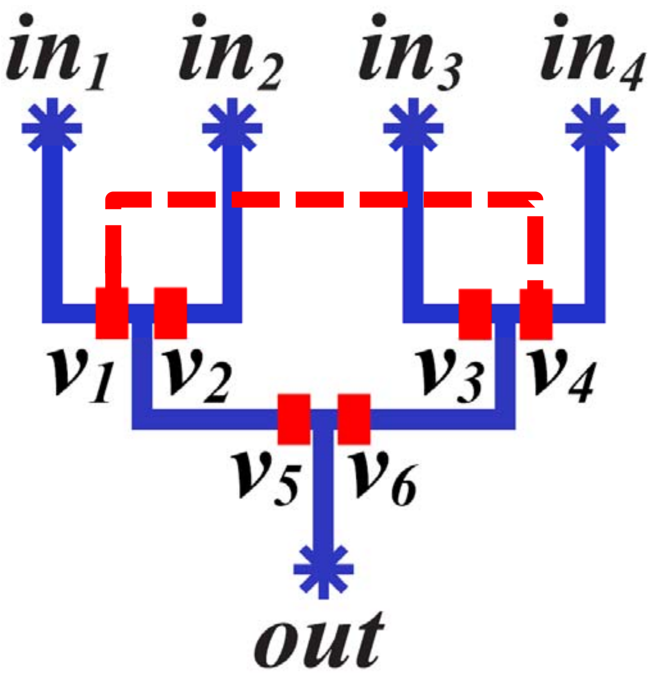
3. Infer Control Sharing



	$in_1 \rightarrow o$	$in_2 \rightarrow o$	$in_3 \rightarrow o$	$in_4 \rightarrow o$
v_1	open	closed		
v_2	closed	open		
v_3			open	closed
v_4			closed	open
v_5	open	open	closed	closed
v_6	closed	closed	open	open

Column Compatibility Problem
- NP-hard
- Reducible to graph coloring

3. Infer Control Sharing

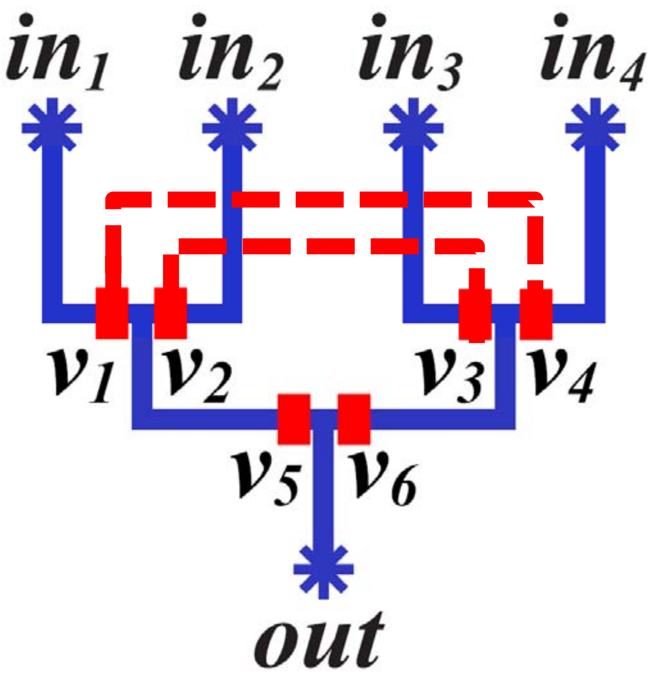


	$in_1 \rightarrow o$	$in_2 \rightarrow o$	$in_3 \rightarrow o$	$in_4 \rightarrow o$
v_1	open	closed		
v_2	closed	open		
v_3			open	closed
v_4			closed	open
v_5	open	open	closed	closed
v_6	closed	closed	open	open

Column Compatibility Problem

- NP-hard
- Reducible to graph coloring

3. Infer Control Sharing

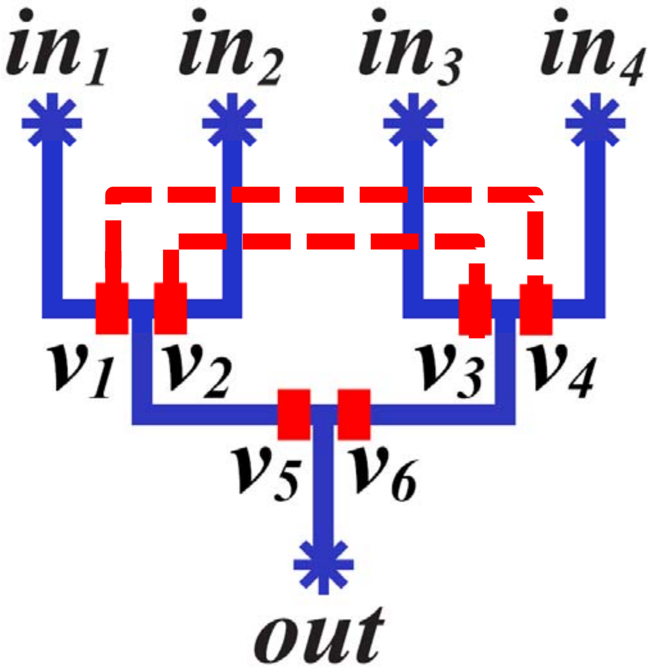


	$in_1 \rightarrow o$	$in_2 \rightarrow o$	$in_3 \rightarrow o$	$in_4 \rightarrow o$
v_1	open	closed		
v_2	closed	open		
v_3			open	closed
v_4			closed	open
v_5	open	open	closed	closed
v_6	closed	closed	open	open

Column Compatibility Problem

- NP-hard
- Reducible to graph coloring

3. Infer Control Sharing

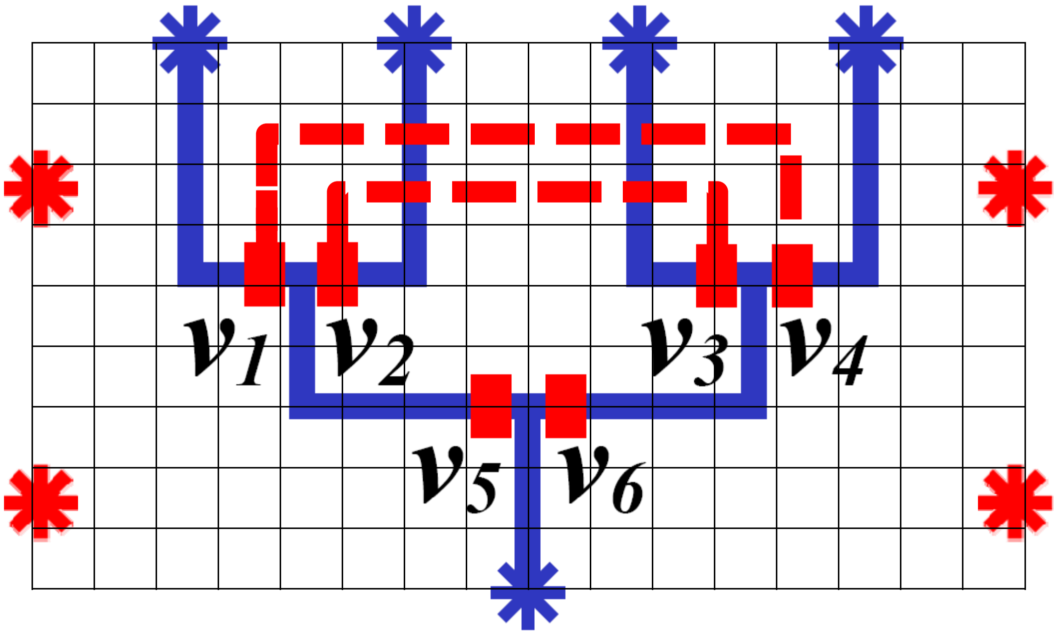


	$in_1 \rightarrow o$	$in_2 \rightarrow o$	$in_3 \rightarrow o$	$in_4 \rightarrow o$
v_1	open	closed		
v_2	closed	open		
v_3			open	closed
v_4			closed	open
v_5	open	open	closed	closed
v_6	closed	closed	open	open

Column Compatibility Problem

- NP-hard
- Reducible to graph coloring

4. Route Valves to Control Ports



- Build on recent algorithm for simultaneous pin assignment & routing [Xiang et al., 2001]
- Idea: min cost - max flow from valves to ports

- **Our contribution: extend algorithm to allow sharing**
  - Previous capacity constraint on each edge:

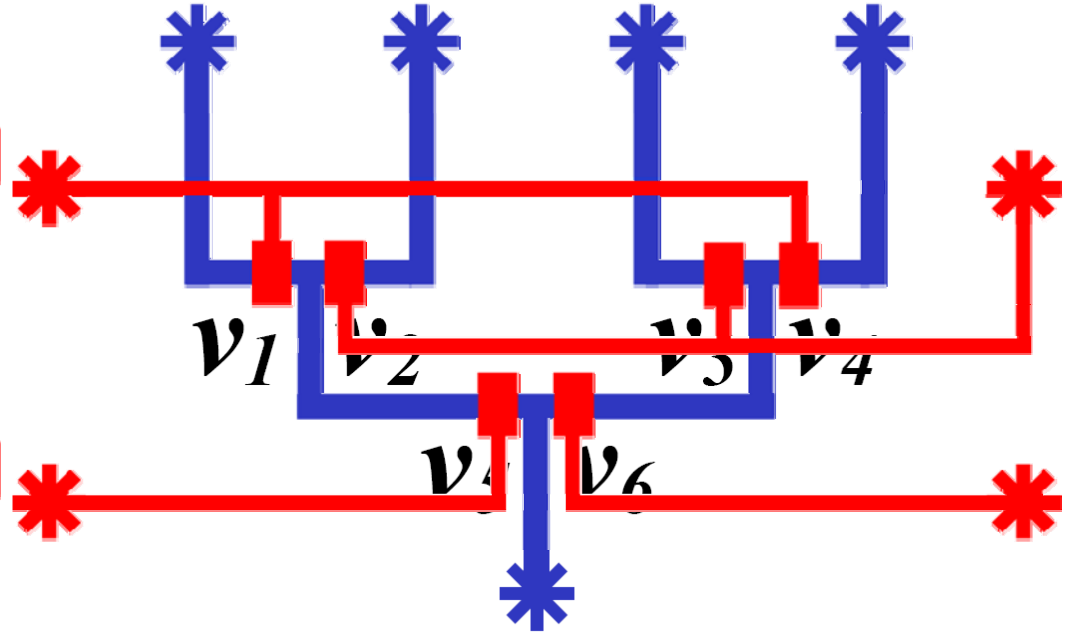
$$f_1 + f_2 + f_3 + f_4 + f_5 + f_6 \leq 1$$

  - Modified capacity constraint on each edge:

$$\mathbf{max}(f_1, f_4) + \mathbf{max}(f_2, f_3) + f_5 + f_6 \leq 1$$

→Solve with linear programming, allowing sharing where beneficial

4. Route Valves to Control Ports



- **Build on recent algorithm for simultaneous pin assignment & routing** [Xiang et al., 2001]
- **Idea: min cost - max flow from valves to ports**

- **Our contribution: extend algorithm to allow sharing**
  - Previous capacity constraint on each edge:

$$f_1 + f_2 + f_3 + f_4 + f_5 + f_6 \leq 1$$

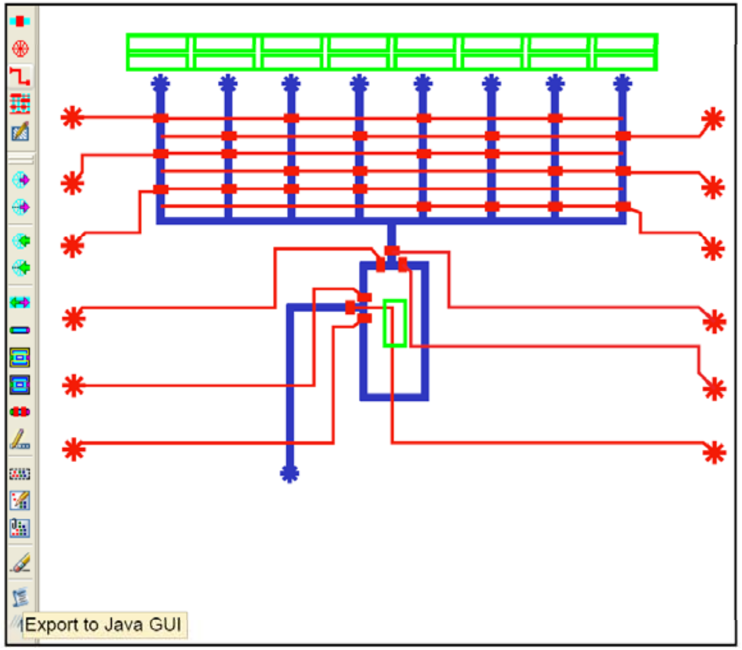
  - Modified capacity constraint on each edge:

$$\mathbf{max}(f_1, f_4) + \mathbf{max}(f_2 , f_3) + f_5 + f_6 \leq 1$$

→Solve with linear programming, allowing sharing where beneficial

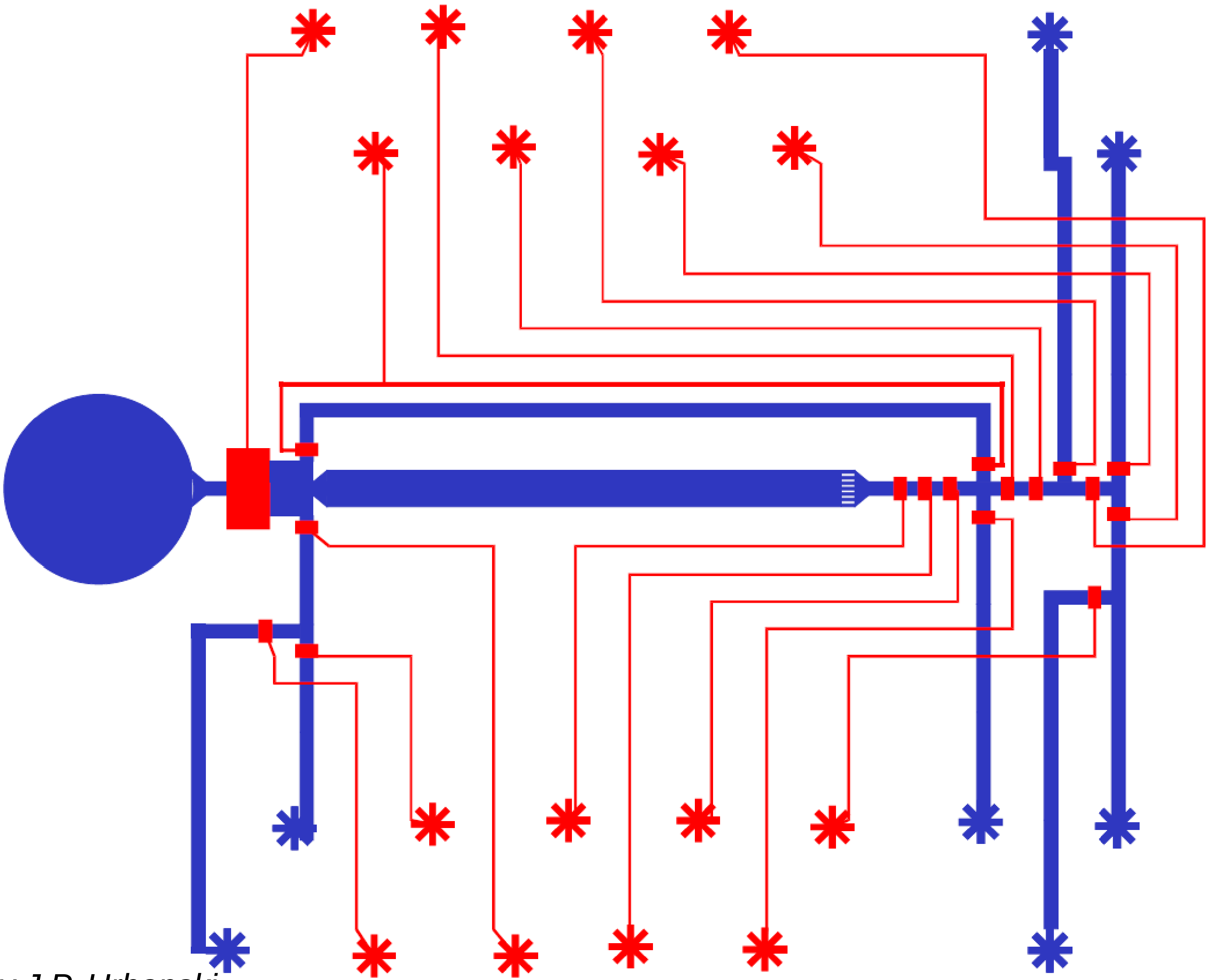
Micado: An AutoCAD Plugin

- **Implements ISA, control inference, routing, GUI export**
  - Using slightly older algorithms than presented here [Amin '08]
  - Parameterized design rules
  - Incremental construction of chips



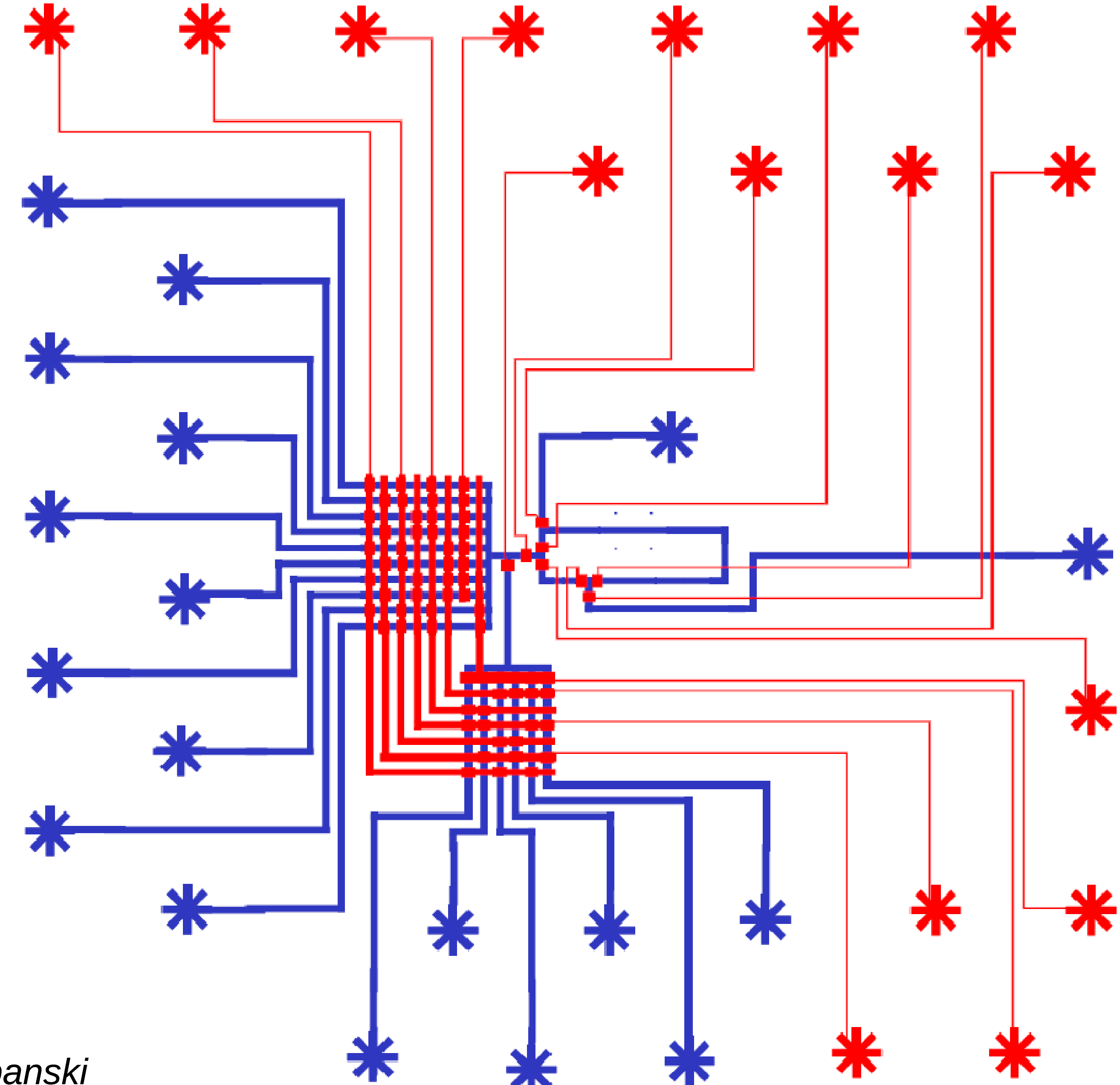
- **Realistic use by at least 3 microfluidic researchers**
- **Freely available at: http://groups.csail.mit.edu/cag/micado/**

Embryonic Cell Culture



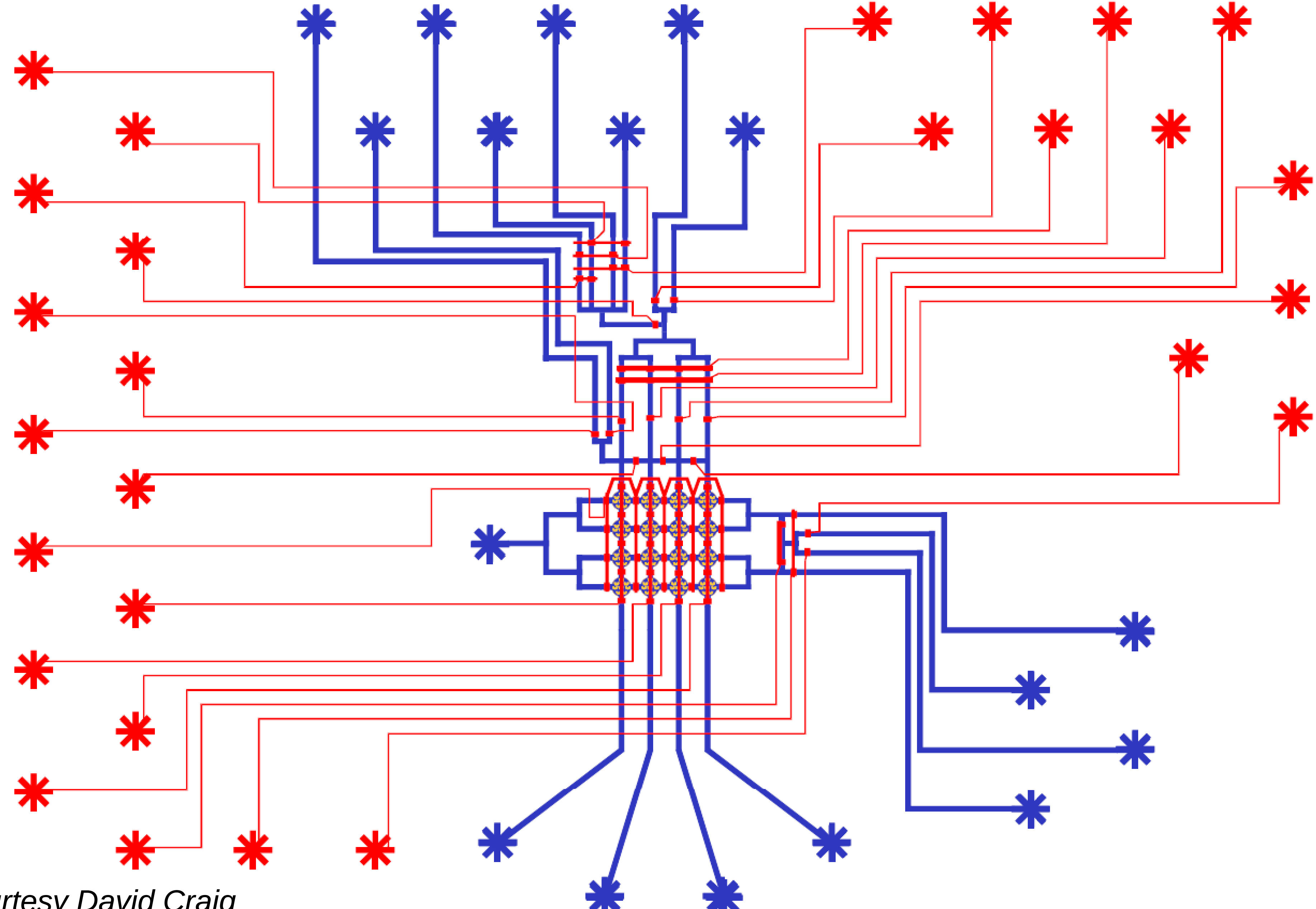
Courtesy J.P. Urbanski

Metabolite Detector



Courtesy J.P. Urbanski

Cell Culture with Waveform Generator



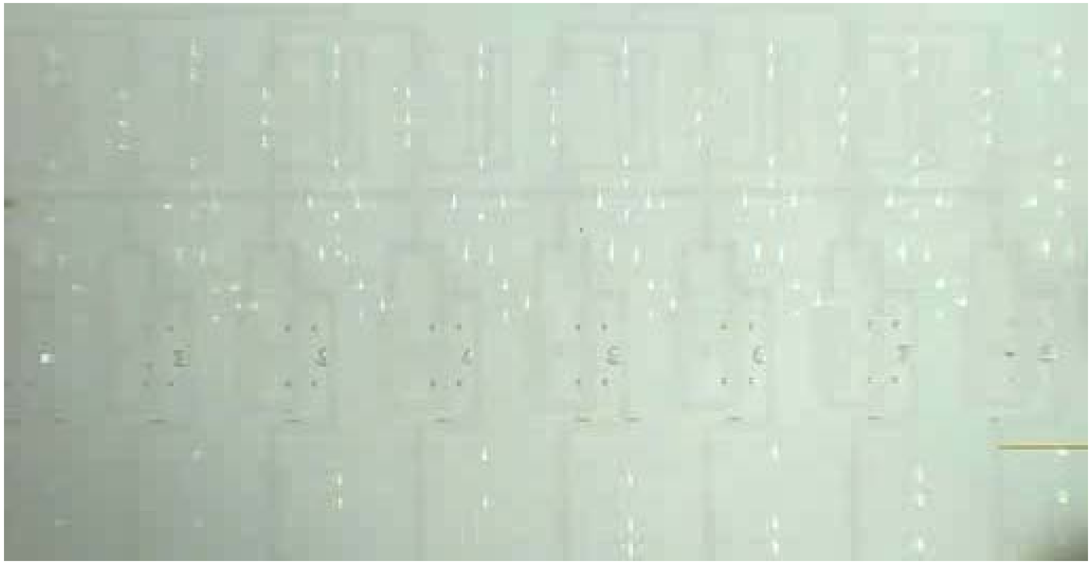
Courtesy David Craig

Open Problems

- **Automate the design of the flow layer**
  - Hardware description language for microfluidics
  - Define parameterized and reusable modules
- **Replicate and pack a primitive as densely as possible**
  - How many cell cultures can you fit on a chip?
- **Support additional primitives and functionality**
  - Metering volumes
  - Sieve valves
  - Alternate mixers
  - Separation primitives
  - …

Conclusions

- **Microfluidics represents a rich new playground for CAD researchers**
- **Two immediate goals:**
  - Enable designs to scale
  - Enable non-experts to design their own chips



Courtesy J.P. Urbanski

- **Micado is a first step towards these goals**
  - Hierarchical ISA for microfluidics
  - Inference and minimization of control logic
  - Routing shared channels to control ports
  - Generation of an interactive GUI

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