

High-Level Programming for Microfluidics

Max Willsey and Jared Roesch
January 12, 2018
Off the Beaten Track at POPL



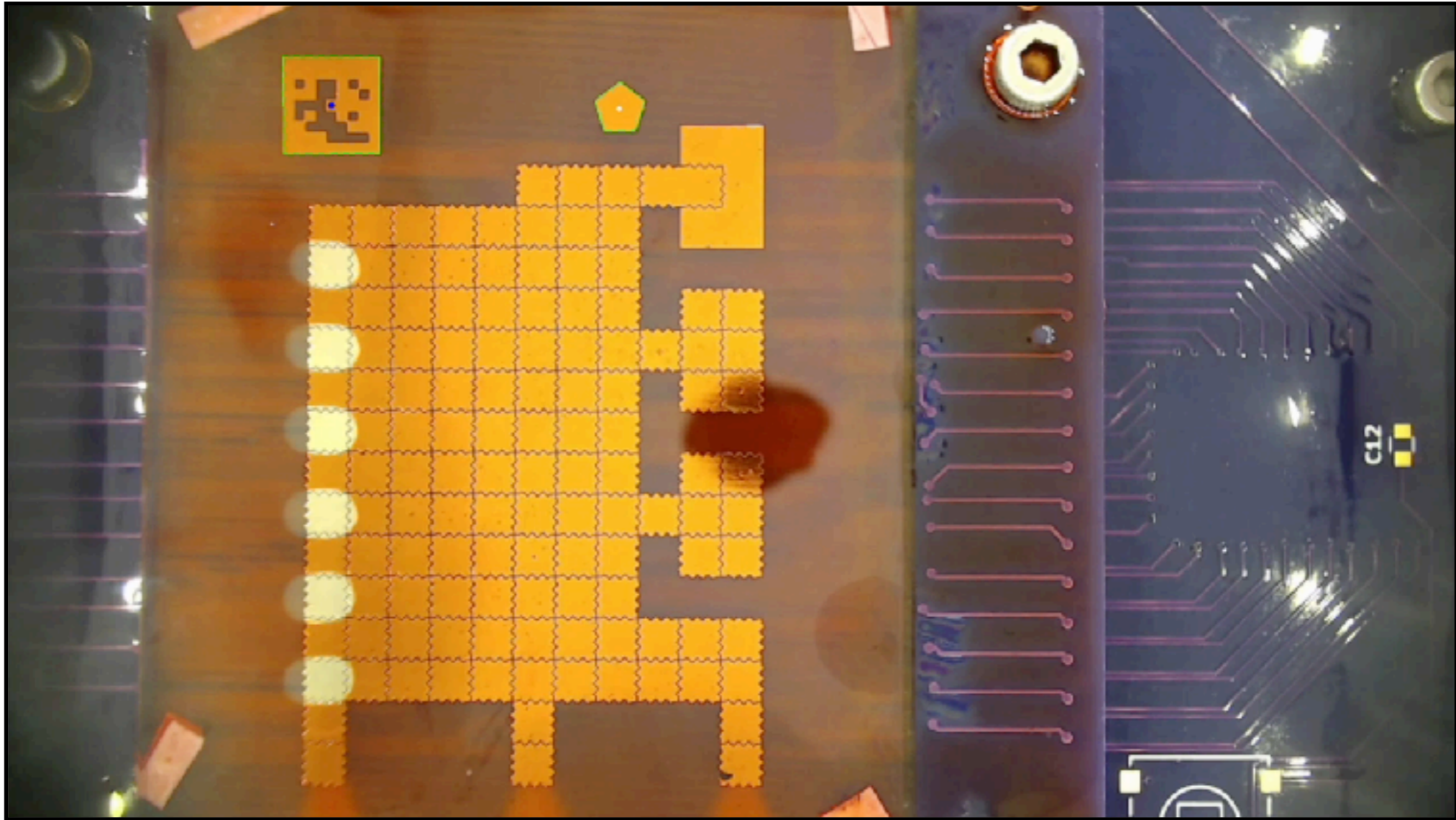
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING



Warning!

Unfinished
(or even started!)
work ahead

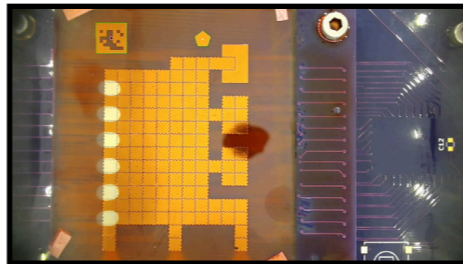
Fluidics?

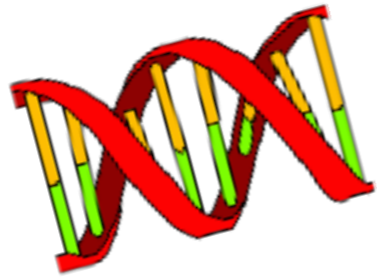


Abstraction Gap

Experiment

Microfluidic Chips





Molecular Computing

Synthetic DNA



Experiment

Chemistry



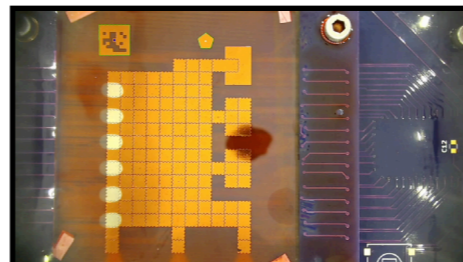
Medical Diagnostics

Medicine

Extensible Fluidic Semantics

Hardware Abstraction

Microfluidic Chips



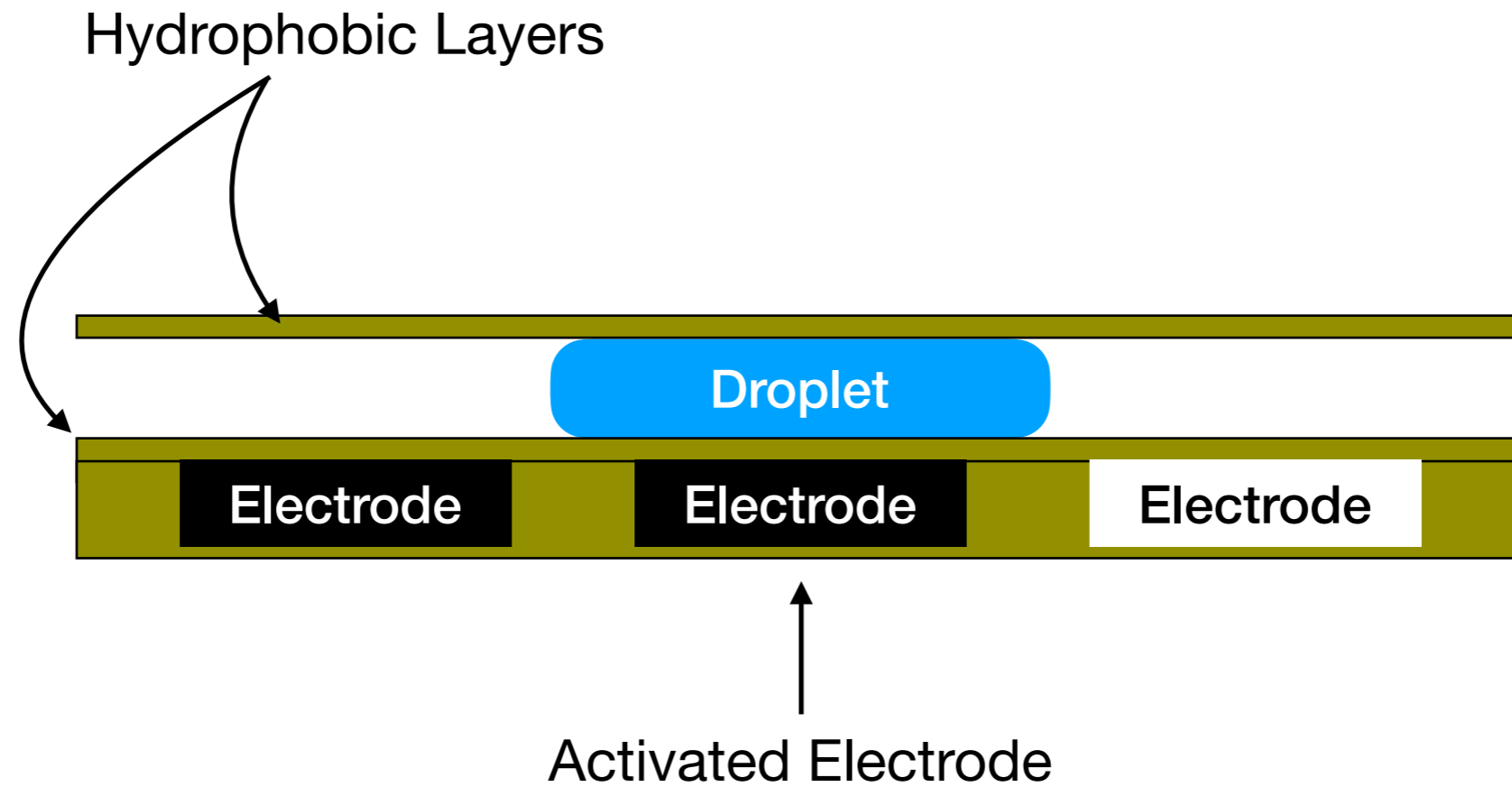
Outline

Extensible Fluidic Semantics

Hardware Abstraction

Microfluidic Chips

Electrowetting



General Purpose Fluidics



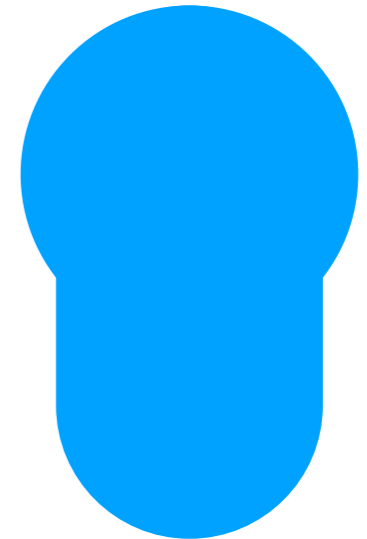
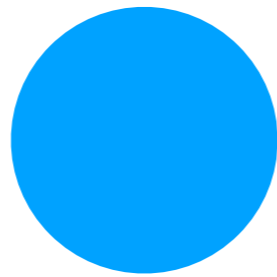
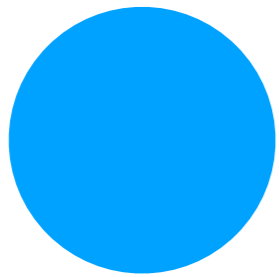
Low Level Programming

```
activate(3,0)  
activate(3,1)  
activate(3,2)  
activate(3,3)  
activate(3,4)
```

```
...
```

```
move_droplet((3,0), (3,4))
```

Error prone



Programming microfluidic devices is hard!

precision

error handling

location tracking

resource management

hardware specific

parallelism

concurrency

domain specific

probabilistic results

**Users are novice
programmers!**

Outline

Extensible Fluidic Semantics

Hardware Abstraction

Microfluidic Chips

What we want

No locations!

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)
```

Automatic error handling!

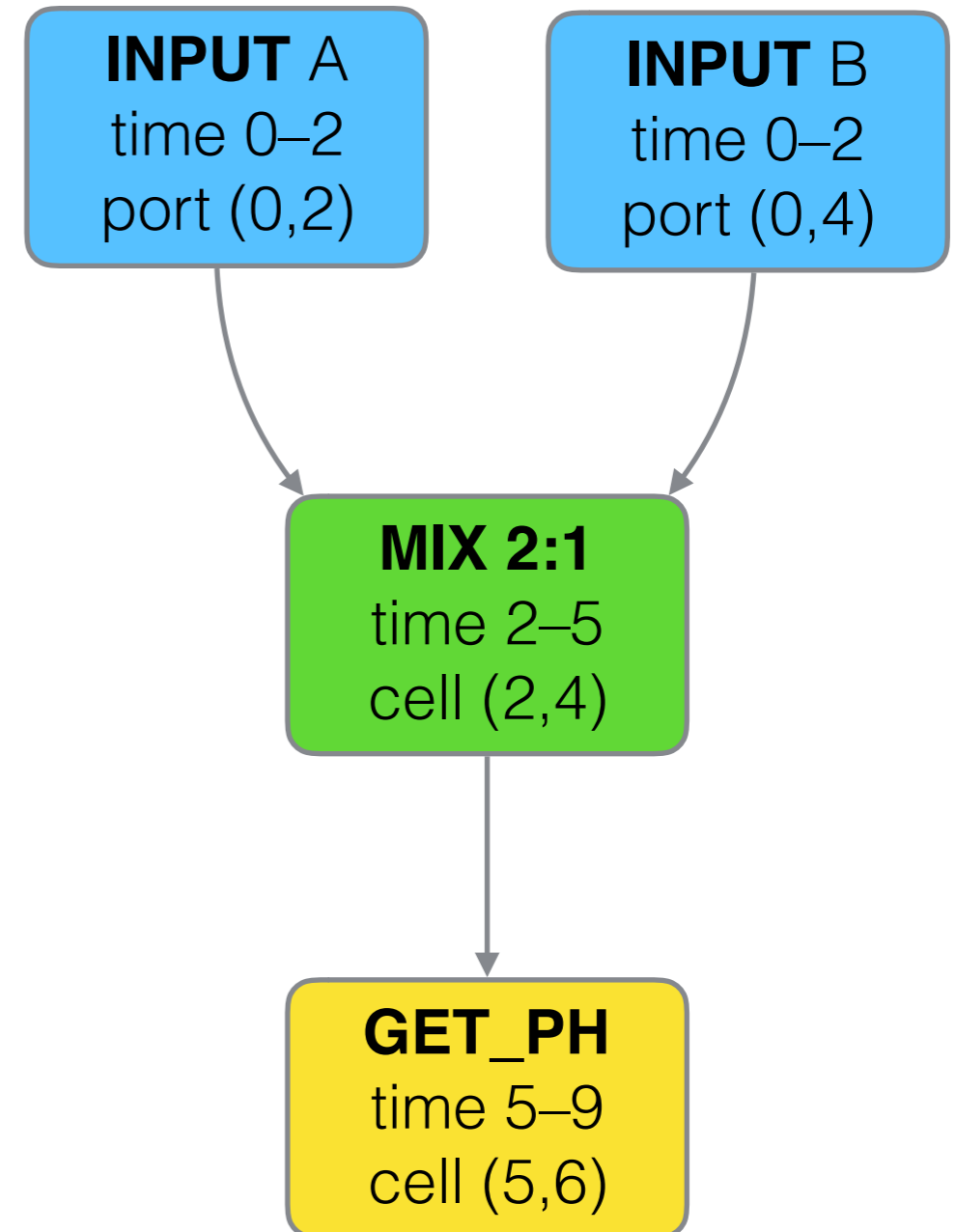
```
while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)
```

Control flow!

No locations

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

get_pH(ab)
```



Dynamism

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)
```

Dynamism

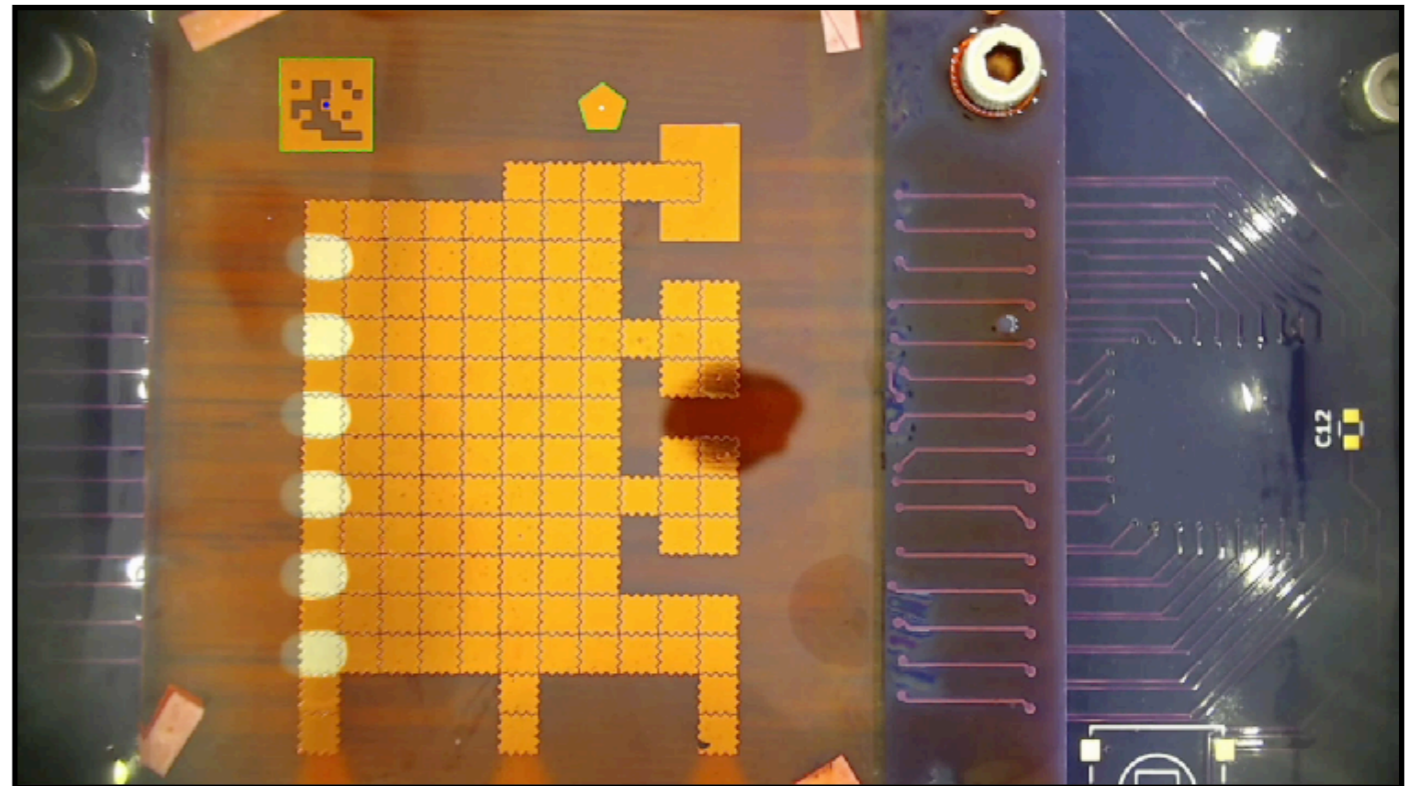
```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)
```

```
while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)
```

data dependent
control flow

Dynamism

**On-the-fly
error correction**



Dynamism

Dynamic error correction

High level programming constructs

**No static reasoning
about resource usage**

Where we are now

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)
```

Outline

Extensible Fluidic Semantics

Hardware Abstraction

Microfluidic Chips

Linearity

```
a = input(substance_A)  
b = input(substance_B)  
# mix in 2:1 ratio  
ab = mix(a, b, 2)
```

```
while get_pH(ab) > 7: }  
    heat(ab)           } long running  
    acidify(ab)        }
```

```
c = input(substance_C)  
ac = mix(a, c)
```

Already consumed!

Linearity

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)
```

```
while get_pH(ab) > 7:           state = Set Droplet
    heat(ab)
    acidify(ab)
```

```
c = input(substance_C)
ac = mix(a, c)
```

Termination?

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)

c = input(substance_C)
ac = mix(a, c)
```

Termination?

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)
```

```
while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)
```

```
c = input(substance_C)
ac = mix(a, c)
```

```
state = Map Droplet {
  ph : Real
}
```

Other Stuff?

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)

c = input(substance_C)
ac = mix(a, c)
```

```
state = Map Droplet {
  ph : Real
  temp : Real
}
```

Other Stuff?

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)

c = input(substance_C)
ac = mix(a, c)
```

```
state = Map Droplet {
  ph : Real
  temp : Real
  volume : Real
}
```

Termination?

```
a = input(substance_A)
b = input(substance_B)
# mix in 2:1 ratio
ab = mix(a, b, 2)

while get_pH(ab) > 7:
    heat(ab)
    acidify(ab)

c = input(substance_C)
ac = mix(a, c)
```

many intrinsic chemical
properties of a sample

procedures,
not primitives

Termination?

```
while get_pH(ab) > 7:  
    heat(ab)  
    acidify(ab)
```

```
@ensures( abs(x.pH - retval) < 0.1 )
```

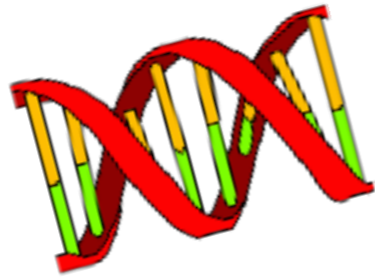
```
def get_pH(x):
```

```
    ...
```

```
@ensures( x.pH - old_x.pH > 0.5 )
```

```
def acidify(x):
```

```
    ...
```

Molecular Computing

Synthetic DNA



Experiment

Chemistry



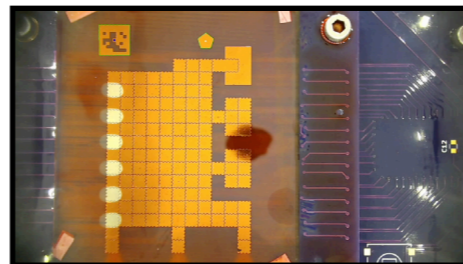
Medical Diagnostics

Medicine

Extensible Fluidic Semantics

Hardware Abstraction

Microfluidic Chips



Thanks!



`misl.cs.washington.edu`

Bonus Slides!

BioCoder

```
for(int i = initial; i <= Threshold; ++i) {
    bioCoder.next_step();
    bioCoder.store_for(tube,94,Time(SECS,45));
    bioCoder.next_step();
    bioCoder.store_for(tube,65,Time(SECS,45));
    bioCoder.next_step();
    bioCoder.measure_fluorescence(tube,Time(SECS,5),"DNASensor");
    bioCoder.IF("DNASensor",GREATER_THAN, .85);
    for(int j = i; j < Total+(Threshold-i); ++j) {
        bioCoder.next_step();
        bioCoder.store_for(tube,94,Time(SECS,45));
        bioCoder.next_step();
        bioCoder.store_for(tube,65,Time(SECS,45));
    }
    bioCoder.next_step();
    bioCoder.drain(tube,"Amplified PCR");
    bioCoder.END_IF();
}
```

Silicon / Molecular Systems

