

Digital microfluidics using soft lithography†

John Paul Urbanski,^a William Thies,^b Christopher Rhodes,^a Saman Amarasinghe^b and Todd Thorsen^{*a}

Received 18th July 2005, Accepted 11th November 2005

First published as an Advance Article on the web 29th November 2005

DOI: 10.1039/b510127a

Although microfluidic chips have demonstrated basic functionality for single applications, performing varied and complex experiments on a single device is still technically challenging. While many groups have implemented control software to drive the pumps, valves, and electrodes used to manipulate fluids in microfluidic devices, a new level of programmability is needed for end users to orchestrate their own unique experiments on a given device. This paper presents an approach for programmable and scalable control of discrete fluid samples in a polydimethylsiloxane (PDMS) microfluidic system using multiphase flows. An immiscible fluid phase is utilized to separate aqueous samples from one another, and a novel “microfluidic latch” is used to precisely align a sample after it has been transported a long distance through the flow channels. To demonstrate the scalability of the approach, this paper introduces a “general-purpose” microfluidic chip containing a rotary mixer and addressable storage cells. The system is general purpose in that all operations on the chip operate in terms of unit-sized aqueous samples; using the underlying mechanisms for sample transport and storage, additional sensors and actuators can be integrated in a scalable manner. A novel high-level software library allows users to specify experiments in terms of variables (*i.e.*, fluids) and operations (*i.e.*, mixes) without the need for detailed knowledge about the underlying device architecture. This research represents a first step to provide a programmable interface to the microfluidic realm, with the aim of enabling a new level of scalability and flexibility for lab-on-a-chip experiments.

Introduction

One of the most exciting scientific developments of recent years has been the progressive miniaturization of chemical and biological instrumentation with an eye towards creating highly integrated “lab-on-a-chip” systems. Compared to industrial laboratory equipment, these systems offer reduced reagent consumption, high throughput and unprecedented automation. Their size makes them suitable for mobile field laboratories, while their low cost allows them to be disposable after sensitive applications. The driving force behind the lab-on-a-chip movement is microfluidics, which enables storage, transport, and manipulation of fluids at the picoliter scale. Microfluidics provides for the experimental sciences what the integrated circuit provided for electronics more than 50 years ago: the ability to integrate diverse micro-scale components in a unified fabrication process. At the device level, microfluidics is becoming a mature technology.^{1–5} Although complex application-driven microfluidic devices have been developed, parallelizing processes such as protein crystallization^{6,7} and proteomic analysis,^{8,9} there has been little progress in the development of scalable and programmable microfluidic systems. Even recent advances in modular microfluidic

breadboards¹⁰ adopt a model where distinct interconnection networks are manufactured for each application. The current user interfaces resemble the earliest days of computing, using software to control individual hardware resources at each time step. Just as programmability was the key to harnessing the power of silicon-based computers, we believe that programmability will likewise be essential for experimental scientists to harness the full power of microfluidic devices.

In this paper, we introduce a platform that enables programmatic control of samples within general-purpose microfluidic chips using immiscible fluids. The general-purpose microfluidic processor contains primitive elements, including storage chambers, input and output ports (I/O), a mixer, and an interconnection network that enables open loop manipulation of samples. As the array of applications continues to expand, such programmable chips will be critical for allowing multiple applications to share hardware resources.

As microfluidic devices continue to evolve, it will be important to have portable applications code that can run without modifications on successive generations of chips. A high-level programming library for microfluidics has been developed which allows the experimentalist to specify general operations (*e.g.*, input, store, and mix) without having to understand the details of the underlying device technology. The mapping from user operations to chip-level logic is specified separately by the chip designer, and could be distributed by a chip vendor with each generation of the device. Mixing and dilution of fluids plays a fundamental role in almost all bioanalytical procedures and, as such, it is critical to provide integrated support for mixing on microfluidic

^aHatsopoulos Microfluids Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA. E-mail: thorsen@mit.edu

^bComputer Science and Artificial Intelligence Laboratory, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA

† Electronic supplementary information (ESI) available: Movies demonstrating the microfluidic latch. See DOI: 10.1039/b510127a

devices. To support flexible mixing, we employ recently developed algorithms for generating a complex mixture using a simple one-to-one mixing device. The algorithm is optimal in that it guarantees the minimal number of mixes to obtain a given mixture.

Our vision is that, by using high-level programs, it will be possible to create large and self-directing experiments that would be too complex to orchestrate at the hardware level. Programmability also offers flexibility, as multiple researchers will be able to replicate an experiment by running the corresponding program on their own microfluidic chip.

Microfluidic implementation

A central challenge in developing a general-purpose microfluidic chip is providing a means to isolate samples of interest and transport them from one location of the chip to another without loss or cross-contamination of the surrounding microchannels. Many approaches have been previously described for on-chip fluid manipulation and separation, implementing techniques such as pressure and electro-osmosis (EO) as the driving force. In microfluidic chips that use a single phase (*e.g.*, a buffered aqueous solution), it is non-trivial to maintain sample integrity. For pressure-driven flows, Taylor dispersion causes dilution of samples during transport as the fluid velocity is faster towards the center of the channel than at the walls.¹¹ Single-phase systems driven by EO do not share this fluid diffusion in straight microchannel flow, as the flow profile is flat rather than parabolic. However, EO-driven flows present different complications, including contamination of charged surfaces driving the flow, high operating voltages and severe fluid dispersion through corner geometries.¹² To overcome the shortcomings of single-phase systems in isolating samples, multiphase systems utilize a discontinuous phase that is emulsified in an immiscible continuous phase, thereby offering an attractive approach to sample confinement and manipulation. As emulsions, the samples do not contact the channel walls directly, preventing cross-contamination, and aliquots of samples may be moved over long channel lengths of any shape while retaining their content.

Digital systems that operate on droplets have made use of multiphase fluidics to freely manipulate individual samples. In several lab-on-a-chip applications, droplet manipulation has been demonstrated using electrohydrodynamic (EHD) forces.^{13–15} Systems based on electrowetting^{16–18} and dielectrophoresis^{19,20} have also been used to dispense, transport, merge, and divide droplets. These approaches are not limited to unidirectional serial flow processes, where segregated samples are periodically generated and transported in the same direction through microchannels; rather, they allow droplets to be transported arbitrarily on a microfabricated control grid. Mixing is accomplished by combining two droplets and agitating the result. However, these systems suffer from imprecision in dividing a combined droplet into two droplets of equal volume, thereby limiting the number of mixes that can be performed for a given error tolerance. For example, a recent electrowetting system²¹ exhibits up to 7% variation in the volume of split droplets, leading to a 25% error margin for a three-stage mixing operation. In addition, sample

volumes in droplet-based electrowetting devices are on the order of 0.5 μL ,^{17,18} two orders of magnitude larger than the volume of samples in our soft lithography device (5.7 nL).

In this paper, we demonstrate individual arbitrary control of discrete emulsified samples in a multilayer polydimethylsiloxane (PDMS) microfluidic device. Flow of a continuous oil phase is used to transport aqueous samples from one area of the circuit to another. A key benefit of this system is precise control over the volume of the samples. In contrast to spherical droplet-based systems, each sample occupies an elongated stretch of the flow channel, with a length much longer than the effective diameter. This enables a high level of precision in setting the sample volume, as pneumatically-actuated elastomeric valves are used to measure a given length of the slug during metering, mixing, and splitting. Minor variations in valve closure have a negligible effect on sample volume, as the volumes displaced by valves are small relative to the total emulsion volume. For example, a peristaltic pump utilizing elastomeric valves can isolate a given fluid volume in a one-phase system with $\sim 0.01\%$ variation (~ 0.6 pl standard deviation per 83 pl injection cycle²²).

While PDMS channels have also been programmatically controlled using Braille display pins,²³ such systems have not been utilized for precise control of individual samples. In order to align, store, and mix discrete samples, the present research relies on control valves that are tightly and arbitrarily spaced (as close as 50 μm). However, as Braille display pins are spaced on an inflexible grid with ~ 2.5 mm between each pin, it is not clear if comparable results could be obtained with such a system. In terms of cost, Braille control is comparable to the solenoid actuators used in the present research: a Braille pin (\$10) controls a single valve, while a solenoid actuator (\$40) toggles a microfluidic control channel that may correspond to eight or more valves. Though both hardware systems are equally amenable to a user-friendly software interface, the current research is the first to present a simple programming model for use by experimentalists who are unfamiliar with the underlying device.

Open-loop manipulation of samples is achieved using a “microfluidic latch”, a partially closed elastomeric valve that allows the continuous medium to pass while blocking the flow of samples. The latch enables precise and consistent transport of samples from one location on the chip to another. Using the latch, we demonstrate a general-purpose mixing device under fully automatic control, without any feedback from the chip to the operating software. The device offers precise metering, mixing, and storage of unit-volume emulsified samples. Not restricted to serial flow processes, the device enables multiple operations to be carried out in parallel, such as mixing one sample while storing another.

Results and discussion

We have developed a prototype implementation of a general-purpose multiphase microfluidic chip using multilayer soft lithography and the silicone elastomer polydimethylsiloxane (PDMS; Sylgard 184, Dow Corning).²⁴ As depicted in Fig. 1, the key features include a rotary mixer,^{22,25} a storage array, fluid input and waste ports, and an interconnection network to

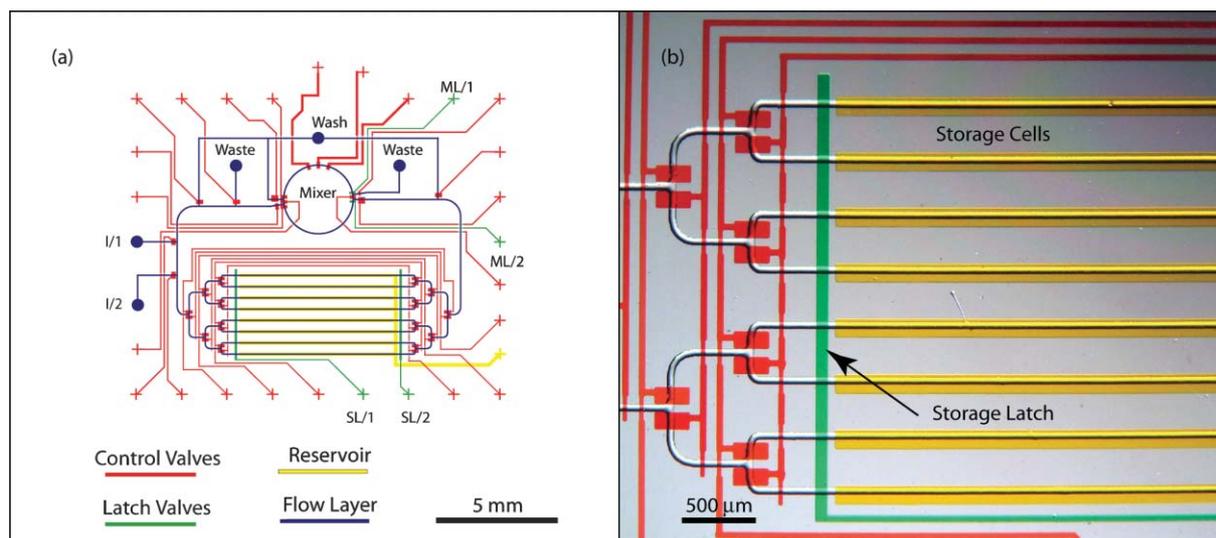


Fig. 1 (a) Schematic of the general purpose microfluidic device. I/1 and I/2 refer to input port for samples 1 and 2. SL and ML refer to storage and mixer latches, and the continuous phase is introduced at the wash port. The software interface automatically controls valves and latches on the control channels to manipulate emulsified samples. (b) Photomicrograph detail of the storage cells; channels are filled with food colorings for clarity. The storage latch aligns slugs into storage cells after preparation in the mixer.

manipulate fluids. The mixer combines two emulsified samples of unit volume and produces two mixed samples of unit volume. The inputs to the mixer are drawn from either the storage cells or the fluid input ports; in the latter case, the mixer valves are used to pinch off a precise unit volume. The outputs of the mixer are transported either to the storage cells or disposed using the fluid waste port. Transport of samples is driven by the pressurized immiscible phase, which is input *via* the wash port. Latches are used to catch transported samples at the boundary of the mixer and the storage cells, thereby enabling automatic control without visual feedback from the device. Storage cells are addressed using a microfluidic multiplexer²⁶ and are shielded by a reservoir layer to prevent sample evaporation.

Sample metering

In a soft lithography device, an emulsified aqueous sample can be trimmed precisely to a given length by pinching off two valves at opposite ends of the sample and then draining the pinched volume *via* a distinct path in the microfluidic circuit. In this research, the term “slug” is used to describe an aqueous sample, with length much longer than effective diameter. Although slugs press against walls of the microchannel, a thin film of continuous immiscible fluid prevents direct contact.²⁷ While fluid metering in one-phase systems has been demonstrated using calibrated peristaltic pumping,²² our focus is the consistent metering of slugs in a multiphase system.

In our device, slug metering is accomplished using the rotary mixer. While a dedicated metering element could also be introduced on the chip, the mixer naturally supports the needed functionality, and each half of the mixer naturally defines a single unit volume under our mode of operation. Fig. 2 illustrates the sequence of operations needed to meter a unit sample into the top half of the mixer. Prior to metering, all microchannels on the chip are filled with the continuous

immiscible phase. In the first step of metering, the aqueous sample of interest is introduced *via* a pressurized input port and directed (using elastomeric valves) to flow through the top half of the mixer to a waste port; this causes the sample to safely overlap the target region. In the second step, the sample is *compacted* against the right hand side of the mixer. This is accomplished by closing the valves on the input port and on the right of the mixer, while opening a path to the pressurized immiscible phase on the upper left of the device. The pressure serves to increase the effective throat diameter of the sample, thereby decreasing its length. A steady state is reached when further compaction becomes negligible at the given operating pressures, and the bulk of the oil between the sample and the channel walls is expelled.

The compaction process is essential for ensuring that metered slugs have a consistent cross-sectional area (and thus a consistent volume). When samples flow directly from the inputs into a system which already contains an immiscible medium, they remain near the center of the channels and the effective throat diameter of the emulsion is small relative to that of the channel. Similarly, as a slug sits stationary in a storage cell, its diameter decreases (and its length increases) in a process termed striation. In order to measure out a consistent slug volume, as well as to compensate for slug striation, slugs are compacted whenever they are transported to a given location. Compaction also helps to reduce slug breakup in the system. As slugs experience a slight elongation during transport, starting with wider slugs helps to mitigate the fluctuations and pinch-off that lead to breakup.

Following compaction, the final step of the metering process is to flush the excess sample from either side of the mixer. As shown in Fig. 2c, the excess sample is transported to the waste port *via* the bottom of the mixer. The result (Fig. 2d) is a precise unit-volume aqueous sample that can be mixed with an additional sample or transported to a storage bin. A

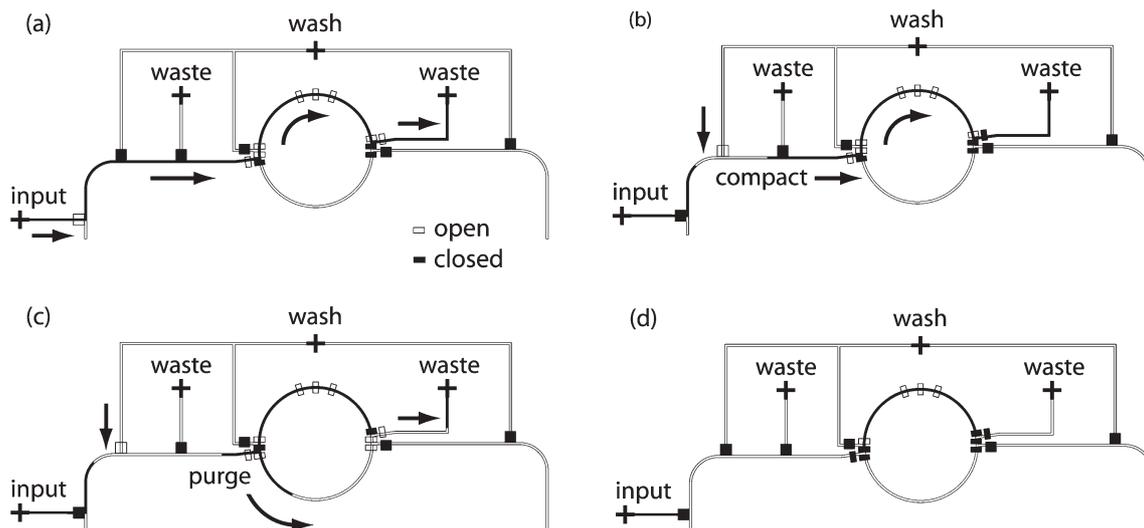


Fig. 2 Microfluidic metering process. Open and closed symbols refer to open and actuated control valves, respectively. (a) Sample of interest flows from an input port through one half of the rotary mixer. (b) Sample is compacted against a valve on the right side of the mixer, ensuring a consistent cross-sectional area. (c) Excess sample is flushed to the waste port. (d) A unit-sized sample results and can be mixed or transported to storage.

symmetrical metering process can also be used to load an input sample into the bottom half of the mixer.

Digital mixing

The on-chip mixer inputs two unit-volume samples, metered as described above, and outputs two unit-volume samples representing the uniform mixture of the inputs. Mixing is performed using a rotary mixer,^{22,25} in which fluids are pumped rapidly around a circuit to accelerate mixing; three integrated elastomeric valves operating in sequence form a peristaltic pump that drives flow. The associated control channels used for high speed peristaltic pumping are located on the top half of the mixer. With the pumps operating at 70 Hz, complete mixing is achieved in approximately 30 s. The lengths of the pump control channels are arranged to have equal path lengths so that time delays due to fluid capacitance, between solenoid actuation and the complete closure of the valves, are equal.

Because our device also uses the mixer for sample metering, it is possible that the microchannels serving as the inputs and outputs to the mixer will be filled with excess sample from a preceding metering operation. To avoid contamination while draining the mixer, we introduce an additional set of input and output channels. As illustrated in Fig. 3, there are four ports on the mixer: the skewed ports are used for loading, while the horizontal ports are used for unloading. Under this configuration, the mixer can be used to meter two separate input samples, mix them to uniformity, and transfer two discrete output samples into separate storage cells.

Microfluidic latch

To align slugs with a given location on a chip (*i.e.*, the mixer or storage cells), we introduce the concept of a microfluidic latch. A latch utilizes a partially deflected multilayer soft lithography valve²⁴ to block the flow of emulsified aqueous samples while

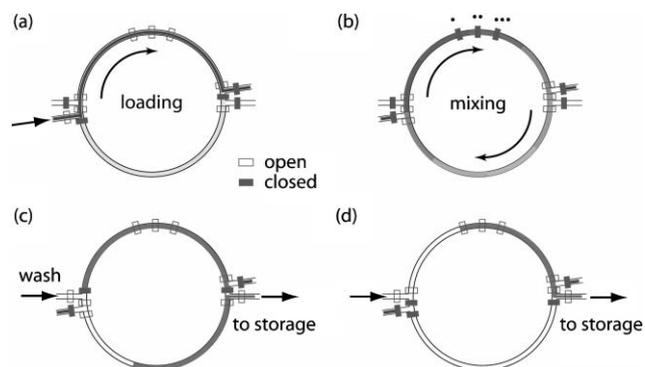


Fig. 3 Schematic of sample loading, mixing and purging within the fluidic processor. Open and closed symbols refer to open and actuated control valves, respectively. (a) The immiscible purge is used to load an aqueous sample, using a valve as a stop. Two segments of fluid, each equal to half of the ring volume, are metered and isolated using the valves on the top and bottom of the mixer. (b) Binary mixing of the sample is completed by peristaltic pumps on the ring periphery operating in a six-step sequence.²⁵ (c) Emulsions in the bottom and (d) the top of the mixing ring are transported to storage cells using the immiscible purge.

allowing the continuous oil phase to pass. By actuating a latch at a given location in the channel and waiting until a moving sample would have safely passed that location, one can consistently align a sample to the actuated latch (Fig. 4). The term “latch” follows from the analogous 1-bit electrical storage element that holds a digital input signal at the output when an external enable switch is true.

This partially closed microfluidic latch catches and retains emulsified aqueous samples, yet allows the immiscible phase of mineral oil to continue to flow along the hydrophobic PDMS surfaces (movies demonstrating the latch are available as ESI†). Surface tension of the colloid prevents the slug from traveling through the microchannel constriction, and the

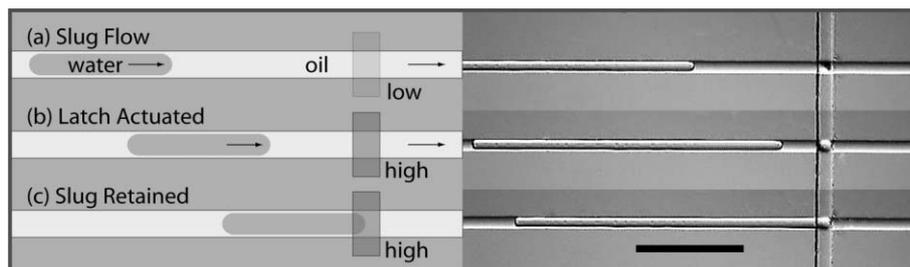


Fig. 4 Schematic diagrams and micrographs of the microfluidic latch used to align samples. (a) The slug is transported by a stream of pressurized oil. (b) After latch actuation, oil continues to flow through the constricted channel but the sample is stopped. (c) Surface tension prevents the sample from deforming. The slug resumes movement once the latch pressure is released. Time between successive images is approximately one second. The scale bar indicates 500 μm .

sample does not break.²⁷ The latch is actuated on a pressure source independent from the control lines so that the channel opening may be fine tuned to catch samples for a given oil pressure.

Fig. 5 illustrates the behavior of slugs under various latching pressures. If the latch pressure is too low, samples immediately break through the latch (the retention time is too small). Conversely, if the latch pressure is too high, oil flow is stopped and the sample is unable to reach the latch (the travel time is too high). The permissible operating range for a latch is defined as those pressures which yield a low travel time while maintaining a high retention time. For the device characterized in Fig. 5, the robust operating range is approximately 14 kPa wide. For any pressure in this range, the latch retains samples upwards of one minute without arresting their transport to the latch. The robustness of the latch is due to the distinct responses of the travel and retention times to increasing latch pressures. Though the slug travel time increases gradually (the

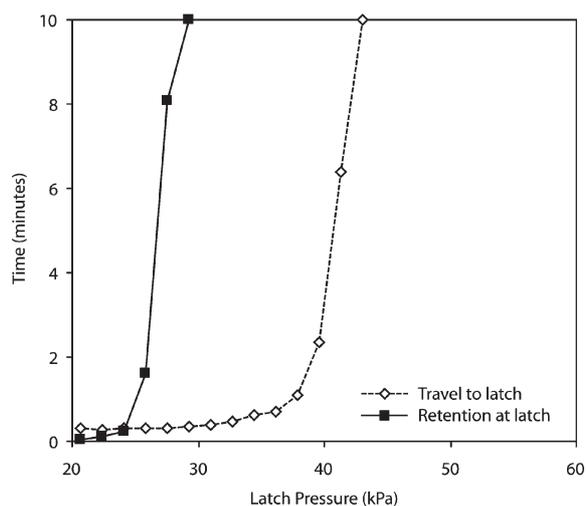


Fig. 5 Effect of latch pressure on travel time (*i.e.*, time required for a slug to travel a 2 mm distance to an actuated latch) and retention time (*i.e.*, time that a slug is retained at an actuated latch without breaking through). At latch pressures >30 kPa, the latch retained slugs for the entire ten minute observation period. Latch pressures between 26 and 40 kPa represent a valid operating range in which slugs are retained without unduly affecting travel times. All measurements were done using a 50 μm flow channel, a 50 μm latch width, 10 μm channel height and a 14 kPa pressure for the continuous phase.

flow rate of the continuous phase is proportional to the latch restriction²⁸), the retention time exhibits a sharp increase after a given threshold. This threshold effect is due to geometric and surface tension properties that inhibit leakage of an emulsion through a latch; once a requisite latch pressure is applied, the slug is retained.

Latches have robust operating points across a range of device geometries and operating parameters. Fig. 6 illustrates preferred latching pressures for two devices, each containing a different flow channel width and a 150 μm latch. A robust latch pressure is determined for various pressures of the continuous phase. The operating points exhibit a simple linear relationship, suggesting that the pressure difference between the latch and oil channels governs the latching behavior.

The latch provides novel, open loop control of emulsified samples that is unique to this system. Thus constant pressure oil-driven aqueous samples may be aligned using open loop timing control to important locations within the device, such as storage or mixer boundaries, without the complications of external sensors or feedback. These surface tension valves are capable of retaining an emulsion for several minutes once it has been stopped, allowing generous tolerance when automatically timing a series of operations. While other researches have developed static Y-junction microvalves to catch and hold liquid streams using surface tension against tapered channel walls, samples may only be released when an adjacent fluid stream overcomes surface tension and subsequently combines with the sample.^{29,30} Other static resistance features exist for catching samples in microfluidic serial processes using hydrophobic patches,¹⁵ or temperature control methods.³¹ However, latches are the most aptly suited alignment mechanism for our application. Latches are straightforward to implement in multilayer soft-lithography systems, as they are operated in the same manner as existing control valves, and no selective surface modification of the PDMS is required. The microfluidic latch is capable of digitally aligning or releasing individual slugs without the need for intermediate fluid streams that may affect internal concentration of samples. While the term “latch” has been previously used by others to describe a phase change microvalve that completely seals a

‡ While external sensors are not needed during operation, we nonetheless calibrate each chip (prior to operation) to determine the best operating pressures and timing parameters.

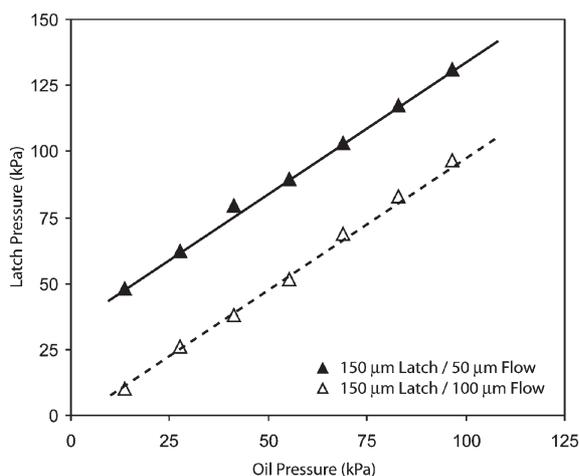


Fig. 6 Preferred latch pressures for two devices across various pressures of the continuous phase. The two devices represent distinct geometries of the flow channel (either 50 μm or 100 μm) and a constant latch width of 150 μm . An effective operating point is found for the latch across all geometries and oil pressures. Reference lines indicate a slope of 1.0.

microchannel when engaged,³² our platform is fundamentally different: it introduces a latch mechanism that allows a continuous phase to pass, yet catches and retains emulsified samples.

Sample evaporation

Evaporation of micro- and picoliter scale aqueous samples in vapor-permeable PDMS devices is an important consideration in the development of a programmable microfluidic platform, as stored samples must remain intact and be accessible at later times. Several approaches to evaporation control have been previously employed. The use of surfactants in a continuous oil phase reduces the rate of water evaporation from emulsions during the self-assembly of colloidal structures.³³ However, the addition of such surfactants is undesirable since it reduces interfacial surface tension and promotes slug breakup during transport. For this reason, surfactants are not used in our system.

To minimize evaporation, our system utilizes a water-filled dialysis reservoir. The reservoir system is based on a previously reported design in which water-filled channels served to reduce sample evaporation in adjacent channels.³⁴ The reservoir is located above the storage area of a microfluidic device, separated from the flow layer by a thin PDMS membrane. The reservoir consists of a set of microfluidic channels that are back-filled with water. The water saturates the surrounding PDMS and decreases the driving potential for evaporation. No noticeable evaporation was observed using the reservoir system (monitored by change in slug length) over periods extending up to three hours.

Device operation

To demonstrate autonomous operation of the device, a series of procedures was completed using open loop control. Food colorings (McCormick) were used as samples, generating and

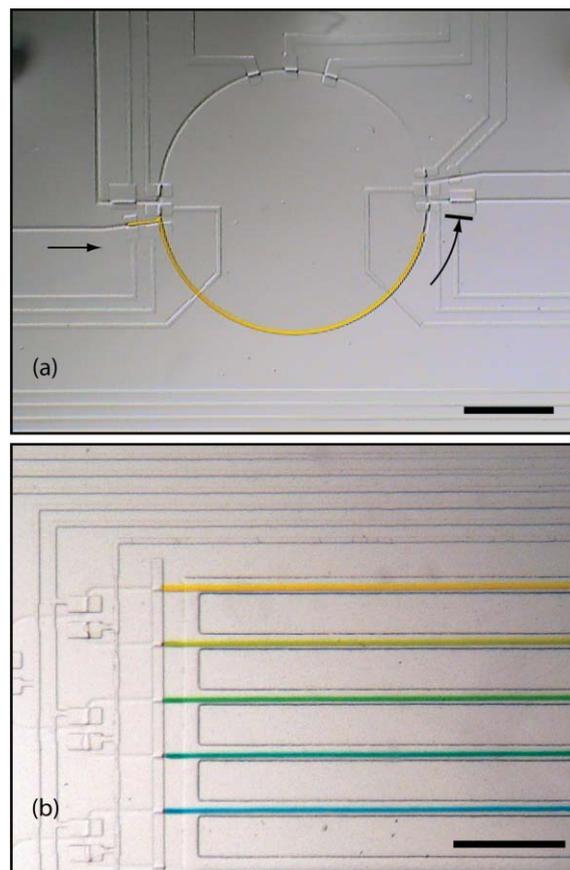


Fig. 7 Procedures are automated using open loop control to demonstrate device operation. (a) Transfer of a unit sample from storage into the mixer. (b) A gradient concentration of two inputs is automatically generated. Ratio of yellow to blue from top to bottom: 1 : 0, 0.75 : 0.25, 0.5 : 0.5, 0.25 : 0.75, 0 : 1. Scale bars indicate 1 mm.

transporting mixtures of yellow and blue inputs. By mixing a blue and yellow sample, a green sample is obtained; by further mixing the green sample with either blue or yellow, a smooth gradient results. The outcome of this process is summarized in Fig. 7.

Programming model and implementation

With the development of scalable mixing, storage and latching primitives, as well as a robust technique for manipulating samples within the microfluidic processor, it is possible to automate any series of operations using software control. Microfluidic valves are externally controlled *via* a Java interface,[§] allowing complex algorithms to be implemented on the device. Any series of mixing operations may be completed without the need for user intervention.

For complex microfluidic chips to achieve widespread adoption, it will be important to present a simple user interface that allows scientists to design and execute experiments without having to understand the details of the underlying device architecture. Because experiments will often run for

[§] The software (and source code) is freely available. For details, visit <http://cag.csail.mit.edu/biostream/>

hours or days at a time, a purely interactive interface is insufficient. Instead, users will need to describe experiments in a simple language that can be automatically executed on the device.

An important criterion for such an experiment description language is that it be *portable*. An experiment description is portable if it contains all the logical information needed to execute the experiment, regardless of the details of the microfluidic architecture. While an experiment relies on certain capabilities of the underlying architecture (ability to mix, transport, store, *etc.*), it should not depend on the specific implementation of those capabilities (rotary mixing *vs.* diffusion, transport as droplets or slugs, number of storage cells, number of valves, *etc.*). A portable language enables the same experiment description to run on multiple generations of microfluidic chips. Similarly, multiple researchers can replicate a given experiment by running the corresponding program on their own particular microfluidic device. Portable programs are also easier to write, modify, and understand, as they describe the logical, high-level operations rather than the mapping of those operations to any particular substrate.

In this research, a portable system for microfluidics programming has been developed based on a Java library. The library conceals all the details of the underlying device architecture and also provides new functions to simplify the process of writing experiments. For example, the following code excerpt produces a concentration gradient between blue and yellow input fluids:

```
Fluid blue = input(1);
Fluid yellow = input(2);
Fluid[] gradient = new Fluid[5];
for (int i=0; i<=4; i++) {
    gradient[i] = mix(blue, yellow, i/4.0, 1-i/4.0);
}
```

To execute the code, a module within the library specifies how to translate each `input()` and `mix()` operation to a sequence of valve actuations on our device. This translation from operations to valve actuations is specified by the microfluidic chip designer, not by the user. If the user obtains a new device in the future, he would also obtain a new library implementation from the chip designer, specifying how to implement each operation on the new device.

The code excerpt also illustrates two interesting features that simplify the task of programming an experiment. The first feature is fluid variables. A variable of type `Fluid` represents a sample with a given composition on the chip. In the programming environment, such variables can be stored and passed to helper functions, just like normal variables. The second feature is the `mix()` operation. A call to `mix (fluid1, fluid2, x1, x2)` returns a new fluid that is the result of mixing `fluid1` and `fluid2` in a ratio of `x1:x2`.

Fluid variables. A key feature provided by the developed software is the use of fluid variables in the programming language. The challenge in implementing this functionality is that physical fluids can be used only once, as they are consumed in mixtures and reactions. However, the programmer might reference a fluid variable (such as “blue” in the example above) multiple times during execution. This

behavior is supported by keeping track of how each fluid was generated and automatically regenerating fluids that are used multiple times.

The regeneration mechanism works by associating each fluid object with the name and arguments of the function that created it. A valid bit is maintained for each fluid, which indicates whether or not the sample is stored in a storage chamber on the chip. By default, the bit is true when the fluid is first created, and it is invalidated when the fluid is used in a mix operation (or other on-chip primitive). If a mix is performed with an invalid fluid, that fluid is regenerated using its history. Note that this regeneration mechanism is fully dynamic; no analysis of the source code is needed.

The computation history created for fluids can be viewed as a tree with several interesting applications. For example, the library can execute a program in a demand-driven fashion by initializing each fluid to an invalid state and only generating it when the user needs the fluid for an experiment. This lazy evaluation affords the library more flexibility in scheduling the mixing operations when the fluids are needed. For example, operations could be reordered to minimize storage requirements or to issue parallel operations. Runtime optimizations such as these are especially promising for the microfluidic domain, as silicon computers operate much faster than their microfluidic counterparts and have cycles to spare at runtime.

Mixing algorithms. The second feature offered by the interface is the mix operation, which combines a set of fluids in arbitrary proportions. This offers significant gains for programmer productivity, as otherwise mixtures need to be synthesized one step at a time using low-level primitives.

To obtain a given mixture on a microfluidic chip, one performs a series of mixes using an on-chip mixing primitive. We adopt a digital model of mixing in which all samples are stored in uniform chambers of unit volume. The mix operation combines two fluids in equal proportions using peristaltic pumping, producing two units of the mixture which can be stored in different storage cells. Using this process, any concentration x/n can be obtained, so long as n is a power of 2. For example, a concentration of 5/8 buffer and 3/8 reagent can be obtained using the mixing sequence illustrated in Fig. 8. However, the programmer should not be concerned with the feasibility of given concentration. In the software, the programmer may specify a concentration tolerance [c_{\min} , c_{\max}] and the system ensures that the mixture produced will fall within the given range. Such error tolerances are already a natural aspect of scientific experiments, as all measuring equipment has a finite precision that is recognized as part of the procedure.

It is natural to suggest a number of optimization problems for mixing. Of particular interest is the number of mixes performed in obtaining a given concentration, as this directly impacts the running time of a laboratory experiment. We have developed a simple algorithm that obtains a concentration using the optimal number of mixes (though not necessarily with the optimal usage of reagents). Though the details of this algorithm are the subject of another publication (in preparation), its operation is fully automatic in the software library.

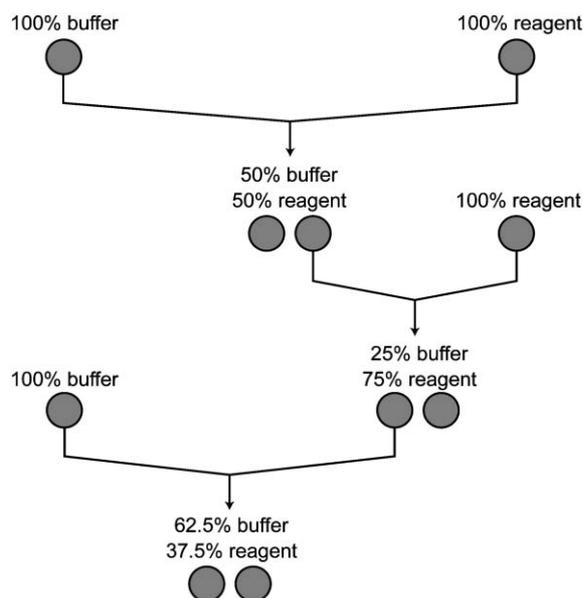


Fig. 8 One possible mixing sequence that yields a concentration of 5/8 buffer and 3/8 reagent. Each mix operation inputs two unit-sized samples and outputs two units of their mixture.

Experimental

Three photoresist-based molds were used to fabricate the multilayer device, corresponding to flow, control, and water reservoir layers patterned with respective microchannel structures. Using standard lithographic procedures, AZ4620 positive photoresist (Clariant) was patterned onto separate silicon wafers using high-resolution design transparencies to create the distinct positive relief molds. To allow complete valve closure in the assembled devices, the channel profiles were rounded on the flow layer mold by placing the wafer on a hotplate at 150 °C for one minute to reflow the photoresist. The post-baked molds were then treated with chloro-trimethylsilane (Aldrich) to facilitate the release of the elastomer during molding.²⁴

Consecutive replica molding from microfabricated masters and chemical bonding steps were used to create three-layer elastomeric devices. The control layer was positioned beneath the flow layer to allow closure of deeper aspect ratio flow channels with the integrated elastomeric valves,²⁸ while the reservoir layer was aligned over the flow layer to control evaporation.

To fabricate the devices, liquid silicone elastomer (Dow Corning Sylgard 184 mixed in a ratio of 10 parts A : 1 part B) was poured on the reservoir pattern mold to a thickness of ~5 mm, and baked at 80 °C for 20 min, while elastomer (4 : 1 A : B) was spun onto the flow master (1000 rpm for 60 s) and baked in a convection oven at 80 °C for ~18 min. The partially cured thick layer was peeled from the master and aligned over the flow mold under a microscope. The two-layer structure was then baked for 20 min, chemically cross-linking the two layers into a single structure. Meanwhile, the control layer wafer was prepared (10 : 1 A : B, 2200 rpm for 60 s) and baked for 18 min. The reservoir and flow layers were aligned to the features of the control layer, followed by another 20 min

bake. To provide mechanical rigidity and to seal the control channels, the device is sealed to a 1 oz glass coverslip coated with a thin layer of PDMS. A featureless No.1 glass coverslip was coated (4000 rpm, 60 s) with a thin layer of elastomer (4 parts A : 1 part B) and baked for 20 min. The bonded flow and control layers were then peeled from the control mold, access ports were punched using a 23 gauge luer stub (BD Biosciences) at both flow, reservoir and control inlets. The microchannels and connection ports were thoroughly cleaned using isopropanol, blown dry with nitrogen, and placed on the partially cured PDMS/glass coverslip. This complete assembly was then baked at 80 °C for a minimum of two hours, bonding the layers into a monolithic device.

In the final assembled devices, the microchannels had the following dimensions: the flow channels were 20 μm (h) × 60 μm (w), the water reservoirs were 20 μm (h) × 200 μm (w), and the control channels were 20 μm (h) × 150 μm (w) in the control valving regions, and 30 μm (w) where they cross under flow lines and valve closure is not desired. The latch valves, also on the control layer, were 20 μm (h) × 100 μm (w). The thin PDMS membrane separating a control channel crossing under a flow channel constitutes a valve; when the pressure in the control channel is high, the silicone membrane is deflected upward into the flow channel and no fluid can pass. Where control channels are narrow, they can cross flow channels without interrupting flow at the valve actuation pressure, as higher pressures are necessary to deflect membranes with smaller surface areas.

Off-chip control of the valves, which are pressurized by a nitrogen source gated by miniature solenoid check valves (The Lee Co.) is carried out using an analog-to-digital logic board (National Instruments PCI-DIO-32HS) driven by the Java operating software. Constant pressure was applied to the reservoirs and regulated by compressed air. Although the device swells slightly due to the use of mineral oil, the operation of microfluidic valves is not affected after saturation of the PDMS. The separate reservoirs containing the oil and aqueous solutions were connected to the microfluidic device through Tygon tubing (500 μm id). Sample and purge inputs were pressurized at 6 psig and 9 psig, respectively. The control channels were initially filled with water to prevent diffusion of air bubbles through the PDMS membrane into the flow layer. The control pressure was maintained at 15 psig, while the upper and lower pressure were set at 11.5 psig and 12 psig respectively.

Light mineral oil was chosen as the continuous phase, as it has a high interfacial tension with water (~0.040 N m), an acceptably low viscosity (~0.022 N s m⁻²), low PDMS swelling properties, and a predictable wetting behavior with channel walls. The continuous phase did not contain any surfactant, facilitating coalescence between samples during mixing.³⁵ To further reduce sample breakup, deep aspect ratio channels were used to increase the cross sectional throat area of the slug and a gentle curvature was placed on all microchannel corners to prevent necking of emulsions during transport.

Conclusions

Microfluidic devices are being embraced by chemists and biologists as powerful tools for performing precise experiments

with high throughput and low cost. This research makes several contributions towards programmable microfluidics. We have presented an emulsion based microfluidic platform and a novel software library that enables high-level control of complex experiments. We present a general-purpose soft lithography microfluidic architecture that provides novel transport of individual samples between fundamental microfluidic components: inputs, outputs, storage chambers, and a mixer. The development of a “microfluidic latch” for catching emulsions within microchannels is a key feature of this approach, and allows open loop automation of this platform. The latch allows the concept of “digital microfluidics” to extend beyond droplet-based architectures into the mature and precise technology of continuous-flow soft lithography.

It will be interesting to evaluate the precision of the soft-lithography mixing device and compare it with programmable EHD droplet systems. To increase the space of potential applications, additional sensors and agitators will be integrated on the device. Scheduling algorithms will be developed to effectively utilize parallel hardware and increase the throughput of experiments.

There are rich opportunities for future work in programmable microfluidics. Using high-level programs, scientists will be able to orchestrate large, adaptive, and reusable procedures that are beyond the grasp of today’s hardware-oriented user interface. As many of the computational sciences have increasingly found overlap with aspects of biology, we are excited that programmable microfluidics might offer a new avenue for programming language and computer architecture researchers to join in the discussion.

Acknowledgements

The authors appreciate helpful conversations with Professor Gareth McKinley, and rheology measurements by Randy Ewoldt. The authors also thank David Wentzlaff and Mats Cooper, who made important contributions to the formative stages of this research. This work was partially supported by National Science Foundation grant # CNS-0305453. J.P.U. was funded in part by the National Science and Engineering Research Council of Canada (PGSM Scholarship).

References

- 1 T. Chovan and A. Guttman, *Trends Biotechnol.*, 2002, **20**, 116–122.
- 2 D. Erickson and D. Q. Li, *Anal. Chim. Acta*, 2004, **507**, 11–26.
- 3 J. W. Hong and S. R. Quake, *Nat. Biotechnol.*, 2003, **21**, 1179–1183.
- 4 S. K. Sia and G. M. Whitesides, *Electrophoresis*, 2003, **24**, 3563–3576.

- 5 B. H. Weigl, R. L. Bardell and C. R. Cabrera, *Adv. Drug Delivery Rev.*, 2003, **55**, 349–377.
- 6 B. Zheng, L. S. Roach and R. F. Ismagilov, *J. Am. Chem. Soc.*, 2003, **125**, 11170–11171.
- 7 C. L. Hansen, E. Skordalakes, J. M. Berger and S. R. Quake, *Proc. Natl. Acad. Sci. U. S. A.*, 2002, **99**, 16531–16536.
- 8 X. X. Chen, H. K. Wu, C. D. Mao and G. M. Whitesides, *Anal. Chem.*, 2002, **74**, 1772–1778.
- 9 J. D. Ramsey, S. C. Jacobson, C. T. Culbertson and J. M. Ramsey, *Anal. Chem.*, 2003, **75**, 3758–3764.
- 10 K. A. Shaikh, K. S. Ryu, E. D. Goluch, J. M. Nam, J. W. Liu, S. Thaxton, T. N. Chiesl, A. E. Barron, Y. Lu, C. A. Mirkin and C. Liu, *Proc. Natl. Acad. Sci. U. S. A.*, 2005, **102**, 9745–9750.
- 11 G. Taylor, *Proc. R. Soc. London, Ser. A*, 1953, **219**, 186–203.
- 12 J. I. Molho, A. E. Herr, B. P. Mosier, J. G. Santiago, T. W. Kenny, R. A. Brennen, G. B. Gordon and B. Mohammadi, *Anal. Chem.*, 2001, **73**, 1350–1360.
- 13 J. Zeng and T. Korsmeyer, *Lab Chip*, 2004, **4**, 265–277.
- 14 O. D. Velev, B. G. Prevo and K. H. Bhatt, *Nature*, 2003, **426**, 515–516.
- 15 M. A. Burns, B. N. Johnson, S. N. Brahmaandra, K. Handique, J. R. Webster, M. Krishnan, T. S. Sammarco, P. M. Man, D. Jones, D. Heldinger, C. H. Mastrangelo and D. T. Burke, *Science*, 1998, **282**, 484–487.
- 16 M. G. Pollack, R. B. Fair and A. D. Shenderov, *Appl. Phys. Lett.*, 2000, **77**, 1725–1726.
- 17 P. Paik, V. K. Pamula and R. B. Fair, *Lab Chip*, 2003, **3**, 253–259.
- 18 A. R. Wheeler, H. Moon, C. A. Bird, R. R. O. Loo, C. J. Kim, J. A. Loo and R. L. Garrell, *Anal. Chem.*, 2005, **77**, 534–540.
- 19 P. R. C. Gascoyne, J. V. Vykoukal, J. A. Schwartz, T. J. Anderson, D. M. Vykoukal, K. W. Current, C. McConaghy, F. F. Becker and C. Andrews, *Lab Chip*, 2004, **4**, 299–309.
- 20 J. A. Schwartz, J. V. Vykoukal and P. R. C. Gascoyne, *Lab Chip*, 2004, **4**, 11–17.
- 21 H. Ren, V. Srinivasn and R. B. Fair, *Transducers*, 2003.
- 22 C. L. Hansen, M. O. A. Sommer and S. R. Quake, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, **101**, 14431–14436.
- 23 W. Gu, X. Y. Zhu, N. Futai, B. S. Cho and S. Takayama, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, **101**, 15861–15866.
- 24 M. A. Unger, H. P. Chou, T. Thorsen, A. Scherer and S. R. Quake, *Science*, 2000, **288**, 113–116.
- 25 H.-P. Chou, M. A. Unger and S. R. Quake, *Biomed. Microdev.*, 2001, **3**, 323–330.
- 26 T. Thorsen, S. J. Maerkl and S. R. Quake, *Science*, 2002, **298**, 580–584.
- 27 J. Bico and D. Quere, *J. Fluid Mech.*, 2002, **467**, 101–127.
- 28 V. Studer, G. Hang, A. Pandolfi, M. Ortiz, W. F. Anderson and S. R. Quake, *J. Appl. Phys.*, 2004, **95**, 393–398.
- 29 J. Melin, N. Roxhed, G. Gimenez, P. Griss, W. van der Wijngaart and G. Stemme, *Sens. Actuators, B*, 2004, **100**, 463–468.
- 30 J. Melin, G. Gimenez, N. Roxhed, W. van der Wijngaart and G. Stemme, *Lab Chip*, 2004, **4**, 214–219.
- 31 J. M. Kohler, T. Henkel, A. Grodrian, T. Kirner, M. Roth, K. Martin and J. Metze, *Chem. Eng. J.*, 2004, **101**, 201–216.
- 32 R. Pal, M. Yang, B. N. Johnson, D. T. Burke and M. A. Burns, *Anal. Chem.*, 2004, **76**, 3740–3748.
- 33 G. R. Yi, T. Thorsen, V. N. Manoharan, M. J. Hwang, S. J. Jeon, D. J. Pine, S. R. Quake and S. M. Yang, *Adv. Mater.*, 2003, **15**, 1300–1304.
- 34 J. Liu, C. Hansen and S. R. Quake, *Anal. Chem.*, 2003, **75**, 4718–4723.
- 35 Y. T. Hu, D. J. Pine and L. G. Leal, *Phys. Fluids*, 2000, **12**, 484–489.