By David W.M. Marr and Toshinori Munakata

MICRO/NANOFLUIDIC COMPUTING

Fluid-based computing may smooth the transition to microscale systems.

Micro/nanofluidic computing is a special-purpose computing paradigm incorporated on small-scale fluidic platforms, which, because of their very small size, have a number of advantages and potential applications in areas such as biomedicine and space technology. The major objective of microfluidic computing is to enhance the functionality of such applications by combining a computing capability with the inherent advantages of microfluidics. Though fluid-based computing does not aim to replace traditional silicon-based technology, computing elements, such as logic gates for logic operations, adders for arithmetic operations, and memory to store information, have been created. To date, most of these have been fabricated at the microscale (hence we use the term "microfluidic" instead of "micro/nanofluidic" for simplicity), but as applications develop, many of these could be further shrunk to significantly smaller, nanofluidic sizes.

ILLUSTRATION BY JEAN-FRANÇOIS PODEVIN

BY COMBINING COMPUTING FUNCTIONS WITH MICROFLUIDICS, MICROFLUIDIC COMPUTING STRIVES TO INHERIT THE ADVANTAGES OF BOTH.

MICROFLUIDICS

Starting around 1990 microfluidic devices have been used [1, 7–9, 11] to control liquids on the micron scale, providing advantages that cannot be realized on the macroscale:

1. Very small quantities are sufficient for analysis and synthesis. For example, with channel heights and widths 1,000 times smaller than macroscopic scales (for example, 1 **μ** vs. 1 mm), the volume is 10^6 times smaller, making the technique especially useful for expensive, hard-to-obtain or hazardous materials.
2. The sizes are close to those of individual cells and molecules. This makes microfluidics more readily capable of manipulating individual cells or DNA molecules, for example.
3. Lab-on-a-chip. Because of the small size, high device densities are feasible. Hundreds or thousands of channels and valves with unique environments can be placed on a small planar surface, allowing for simultaneous parallel and complex analyses. This is analogous to computer integrated circuits where many circuits are integrated on a small silicon chip and where a high degree of parallelism contributes to overall faster processing times.
4. Processing times for individual analysis and syntheses are typically much shorter than equivalent macroscale processes because heat and mass transport and chemical/biological reactions occur over significantly smaller length scales.
5. Items 3 and 4 allow analysis and synthesis to be performed at the point of need rather than a centralized laboratory.
6. Fabrication methods are based on traditional silicon-based technologies, and handling techniques are relatively easy to learn.

RECENT DEVELOPMENTS IN MICRO/NANOFLUIDIC COMPUTING

By combining computing functions with microfluidics, microfluidic computing strives to inherit the advantages of both. The use of fluidic systems for logic at macroscopic-length scales is not new, however, the approach withered as competing electronic devices were scaled down to microscopic and nanoscopic dimensions. These earlier devices were designed by taking advantage of the turbulent flows present at relatively large sizes. Microscale flows, however, are dominated by viscous effects where turbulence is not present (such as when flows are laminar), and microfluidic logic must use different techniques. To address this, various computing elements based on different principles have been implemented, though one characteristic common to each is the need to manipulate fluid in microchannels; beyond this, every implementation is unique. Examples include:

Logic gates by electrochemical reactions. The group of Richard Crooks at Texas A&M University has developed logical OR and NAND gates, analogous to solid-state counterparts, such as diodes and transistors, and based on electrochemical reactions. This approach employs microfluidics as a means of replenishing the necessary reagents but not directly as a means of performing the logical operations themselves [12].

Logic gates by relative resistance. We have implemented basic logic gates, such as NOT, AND, OR, and NAND, within a microfluidic device based on the relative resistance of fluid flows [10]. This approach is purely fluidic, does not require additional electronic control, and hence is relatively simple, inexpensive, and versatile. One unique feature of this implementation is that a number of logic gates are realized simultaneously within the same unit. This not only allows flexible selection of desired gates for a specific application but can also lead to compound operation by combining some of these gates. For example, one can implement a half adder by using the XOR gate to represent a significant bit and the AND gate as a carry bit.

Nanofluidic transistors. The team of Arun Majumdar and Peidong Yang at the University of California, Berkeley, have created a nanofluidic "transistor" that allows them to control the movement of ions through nanoscale, water-filled channels. In this, channels were fabricated between silicon dioxide plates spaced 35nm apart and filled with an aqueous potassium chloride solution. Across this channel, they applied a voltage, shutting off potassium ion flow. This approach is basically a hybrid system of nanofluidics and silicon-based computing; the computing is performed by a traditional chip but now operated upon a fluid system. The flow of ions either exists, representing the "on" state, or is shut off, representing the "off" state, effectively a nanofluidic transistor controlled by a voltage at a gate in the channel. By combining these switches, they should be able to construct logic gates and memory in the future; however, integration is a problem still to be addressed. Despite this, the approach has no moving parts, valves, pumps, and mixers and is therefore simpler than those requiring such components [5]. In addition, the approach can be extended from control of ion transport to larger macroscale molecules, such as proteins [4].

Memory. Though fluid-based logic can form the basis of future computing capabilities, ancillary components, such as memory not based on logic gates, will make this more practical. The group led by Stephen Quake at Stanford University has developed microfluid-based memory elements through the use of viscoelastic polymer solutions. Mimicking functions of solid-state components, they have demonstrated both a flux stabilizer and bistable flip-flop memory by taking advantage of non-linearity in the fluid viscosity [2]. In addition, they have demonstrated memory storage using large-scale microfluidic valving techniques [3].

Bubble logic. More recently, Manu Prakash and Neil Gershenfeld at MIT have implemented logic gates, including AND, OR, and NOT, using bubbles in microfluidic channels [6]. By employing appropriate channel configurations and time lags (on the order of ms) for arrival of individual bubbles, logic gates are realized, leading to a cascadable and scalable computing scheme.

BASIC CHARACTERISTICS AND APPLICATIONS

Although fluidic computing has been implemented in significantly different ways, there are some common characteristics. The speed of fluidic computing is of order 1 sec down to 1 msec, much slower than silicon-based analogues. Hence, this approach will not replace conventional computers; rather microfluidic computing will enhance microtechnology through direct incorporation of computing functions in devices with other primary functions. The goal therefore is to increase the efficiency and accuracy of information transfer and processing within integrated microsystems. As an alternative, a microfluidic system may be connected to a traditional computer for computation (a fluidic-silicon hybrid environment). In this, data is measured and fed into a computer, computation is performed, and then output sent back to the system. Which approach is better, standalone fluidic or hybrid, depends upon many factors. The standalone approach may be better for information transfer, cost, and space requirements, while the hybrid approach could be chosen if extensive and/or complex computations are involved.

Basic types of operations associated with microfluidic computing include: determination of physical/chemical/biological properties in the fluid (such as the detection of specific molecules or cancer cells), logic and arithmetic operations based on the determined results, storing and retrieving of intermediate values, taking actions based on the outcome (for example, mixing flows or dispensing drugs), and interfacing with external systems. The field of microfluidic computing is quite new, and each of these operations will likely be explored in the future, with relatively simple implementations explored before more complex ones. These will involve a low degree of parallelism and simple logic operations; for example, during a microfluidic process, one has an intermediate result, such as the existence of a certain chemical or biological component, or a physical quantity exceeds a threshold value as $x > x^*$. These can be expressed as a Boolean value A = 1. Given other Boolean measures, such as B, C, and so forth, one takes a certain action, D. For example, D = 1 may represent the dispensing of a certain chemical and D = 0 not to dispense. D may be represented as a compound logic expression of A, B, and C, that is, "if (A = 1 AND B = 1 AND C = 1) OR (A = 1 AND at least one of B OR C = 1), then NOT D." This type of expression can be simplified by Boolean algebra, then implemented as microfluidic logic. As in other new computing paradigms, architectural design and algorithm development may play important roles when these operations become more extensive. One common technical issue that needs to be solved for some cases is cascading for microfluidic integrated circuits.

Specific applications for microfluidic computing could include:

- Blood tests, which require very tiny samples for automated multipurpose, parallel processing analyses, with rapid results. Current test-tube scale analyses could become obsolete.
- Drug/chemical development and testing. Each analysis requires very small amounts and is therefore cost effective for expensive samples. Hazardous samples can be handled with less risk. Hundreds of tests can be performed in parallel, leading to faster development times.

REFERENCES

1. Beebe, D.J. et al. Functional hydrogel structures for autonomous flow control inside microfluidic channels. *Nature 404* (2000), 588–590.
2. Groisman, A., Enzelberger, M., and Quake, S.R. Microfluidic memory and control devices. *Science 300* (2003), 955–958.
3. Groisman, A. and Quake, S.R. A microfluidic rectifier: Anisotropic flow resistance at low Reynolds numbers. *Physical Review Letters 92*, 094501 (Mar. 5, 2004).
4. Karnik, R., Castelino, K., and Majumdar, A. Field-effect control of protein transport in a nanofluidic transistor circuit. *Applied Physics Letters 88*, 123114 (2006).
5. Karnik, R. et al. Electrostatic control of ions and molecules in nanofluidic transistors. *Nano Letters 5*, 5 (May 11, 2005), 943–948.
6. Prakash, M. and Gershenfeld, N. Microfluidic bubble logic. *Science 315* (2007), 832–835.
7. Rosamund, D. et al. Insight: Lab on a chip. *Nature 442*, 7101 (2006), 367–418.
8. Terray, A., Oakey, J., and Marr, D.W.M. Microfluidic control using colloidal devices. *Science 296* (2002), 1841–1844.
9. Unger, M.A. et al. Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science 288* (2000), 113–116.
10. Vestad, T., Marr, D.W.M., and Munakata, T. Flow resistance for microfluidic logic operations. *Applied Physics Letters 84*, 25 (June 21, 2004), 5074–5075.
11. Whitesides, G.M. and Stroock, A.D. Flexible methods for microfluidics. *Physics Today 54*, 6 (2001), 43–48.
12. Zhan, W. and Crooks, R.M. Microelectrochemical logic circuits. *J. Am. Chem. Soc. 125*, (2003), 9934–9935.

DAVID W.M. MARR (dmarr@mines.edu) is a professor in the Chemical Engineering Department at Colorado School of Mines.
TOSHINORI MUNAKATA (t.munakata@csuohio.edu) is a professor in the Computer and Information Science Department at Cleveland State University.
