Towards Microelectrofluidic System (MEFS) Computing and Architecture

A. Dewey, R. Fair, J. Jopling, J, Ding, T. Zhang, F. Cao, B. Schreiner, and M. Pollack

Duke University, Dept of Electrical and Computer Engineering
P.O. Box 90291, Durham, NC 27708-0291, jj@ee.duke.edu

ABSTRACT

This paper presents the rationale, design, and simulation of next-generation microelectrofluidic system computational architectures for the emerging field of bioinformatics. Current microelectrofluidic processors (e.g. biochips) have largely dedicated architectures supporting relatively specialized applications. A more general microelectrofluidic system computational architecture is given involving a multi-drop bus, pipelined structure. Functional requirements are explained and results of performance modeling and analysis of the microfluidic processor architecture are presented.

In developing architectural concepts for microfluidics, it is instructive to "map" fluidic computing concepts into electronic computing concepts to gain insight into useful organizational structures. With this linkage, issues of optimal ways to sequence data movement to affect the execution of an instruction apply to the movement of liquid to affect the execution of a protocol. Leveraging the extensive technology base of electronic computing architecture, the organization of a fluidic architecture is presented. Performance modeling and simulation studies are conducted to understand quantitatively issues of fluid operations, resource utilization, and overall application throughput.

Keywords: MEFS, microfluidics, biochips, fluidic architecture

1 INTRODUCTION

Enthusiasm over the potential of microelectrofluidic (MEF) devices has spawned a wealth of research that has in turn produced myriad realizable components and devices. Significant progress has been made in individual MEF components. For example, fluidic channels [8], switches, valves [5], [12], pumps [10], flow meters, joinery/connectivity [4], etc. have been designed, built and studied extensively. As the capabilities advanced and these discrete components improved, capabilities emerged to design and build small systems that combine existing components into a more useful device. For example, devices for genetic diagnostics [6], [9], devices for protein analysis [11], [3], devices for bioassays and immunoassays [1], [2], devices with wide ranging medical potential [7], and more, have all been conceived and many have been built. Simultaneously, a new industry is growing, manufacturing and marketing such devices.

The limitation of present designs, however, is that devices tend to be application/analysis specific. There is no one system capable of performing a collection of differing analyses or procedures. In addition, in a number of systems only the analysis device is microfluidic in nature, while the remaining pre-/postprocessing is performed by normal macroscopic, possibly automated, methods. The potential for a universal architecture is great, one in which multiple, differing procedures may be performed with simple reconfiguration and possibly in parallel. Here is presented a rationale behind movement toward a reconfigurable and reusable computational MEF architecture, a design of such an architecture, and preliminary simulation and performance modeling of this architecture.

2 FUNCTIONAL REQUIREMENTS

The much revered goal of such fludic systems is the much touted *laboratory-on-a-chip*, or *micro-total analysis system* (μTAS). Such a system must be versatile enough to handle a range of procedures for any laboratory or environment in which it would be employed, anywhere from a research laboratory to a crime scene to on-site emergency medical care to the front line of battle. This being the case, it is clear the potential applications are wide ranging.

This is where the use of analogies to the more developed electrical world are most useful. A detailed look at a set of potential applications reveals that despite the wide variety of procedures there exists a fundamental 'instruction set' of fluidic and biochemical operations that can be 'reprogrammed' to accomplish countless larger scale operations (Figure 1). What ultimately makes each procedure unique is the final analysis step. In turn, each of these 'instructions' requires that certain hardware units be present for execution, much the way an ADD operation could not execute without an ALU.

As mentioned above, the system must be fairly flexible in its programmability. This reconfigurability can occur on two levels, either by electrical control or through

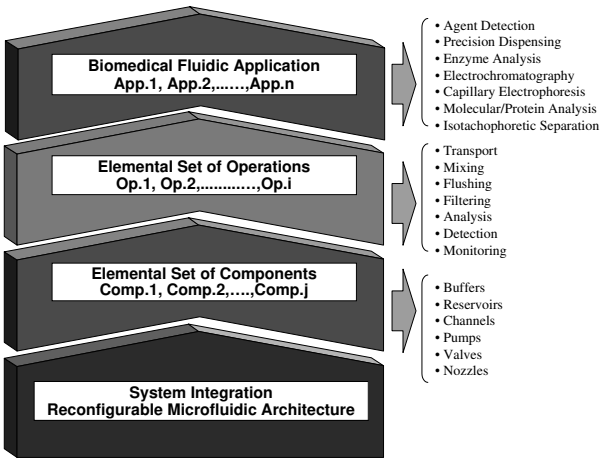

Figure 1: Microelectrofluidic Applications

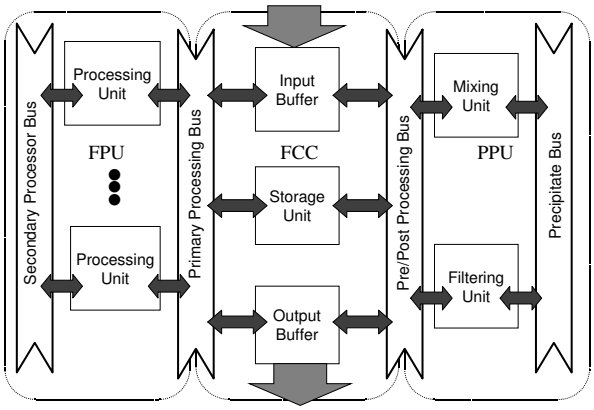

Figure 2: Microfluidic Architecture

physical instantiation. Electrical control is little different from that found in established computational architectures, and is dictated by an overseeing program. Physical instantiation involves modular components that can be removed and added to the architecture to meet a specific demand. This alternative is also being considered through related research on an *intelligent substrate*.

3 ARCHITECTURE

Having identified the component set necessary to realize the instruction set, the question remains as to how to organize and interconnect them to give the best performance. Dealing in a fluidic/chemical domain rather than an electrical domain, the key is the observation that each canonical operation will likely have differing execution times. This can depend on the operation, the overall procedure performed, and the respective fluids involved. The simplest example is the execution of the final analysis step, which will typically last much longer than the preceding preparation steps. In addition, unlike in modern computer architectures, these operations cannot be further broken down into a series of small steps capable of being pipelined. The lowest level of pipeline granularity occurs at the level of the canonical operation.

This observation of disparate execution times implies a bus oriented architecture, where devices can be accessed in parallel with no requirement that one operation completes before another operation initiates. Furthermore, more frequently accessed and faster executing operations will necessitate more frequent use of the bus than comparably longer and infrequent operations.

These factors are reflected in the initial MONARCH v1.0 architecture proposal (Figure 2). The design is one centered around multiple buses. The *Fluidic Central Controller (FCC)* is the central storehouse, providing an interface into and out of the macroscopic world, as well as intermediate storage for running processes. The *Fluidic Processing Unit (FPU)* communicates with the FCC via the *Primary Processing Bus*. Units within the FPU may communicate amongst each other via the *Secondary Processing Bus* without having to occupy the primary bus. In addition, on the opposite side of the FCC, communicating via the *Pre/Post Processing Bus*, is the *Pre/Post Processing Unit (PPU)*, in which additional preparation and cleanup processes may occur. Similarly to the FPU, the units of the PPU may communicate with each other via the *Precipitate Bus*, where they may exchange reagents or expel waste by-products without having to occupy the Pre/Post Processing Bus.

4 PERFORMANCE SIMULATION

Figure 3 shows a petri net representation of the system architecture. The two major buses that connect the FCC, PPU, and FPU together are represented as resources. When a sample enters the system, it can take one of two branches, depending on which resources are available. For example, if the FPU bus is free, the sample will be transfered to storage units through the FPU bus. Once in the storage cell, it will again wait for the needed bus resource to be available. This time it is transfered to the PPU. This process will continue until the sample has finished all processing steps and is sent back to the storage cells in the FCC. There it waits for a bus to be available to be expelled from the chip.

A couple of notes about the model and simulation. First, it is necessary to observe that the peripheral buses are considered internal components of the FPU and PPU and, therefore, not explicitly in the Petri net model. Second, this model assumes a simple procedure that requires only one trip each to the PPU and FPU to complete operation. Input and output, of course, are the duty of the storage buffers in the FCC. This model also assumes that input fluids are first stored before processing occurs, and output fluids are stored before being expelled (as opposed to possibly buffering from

input/output directly to/from a processing unit).

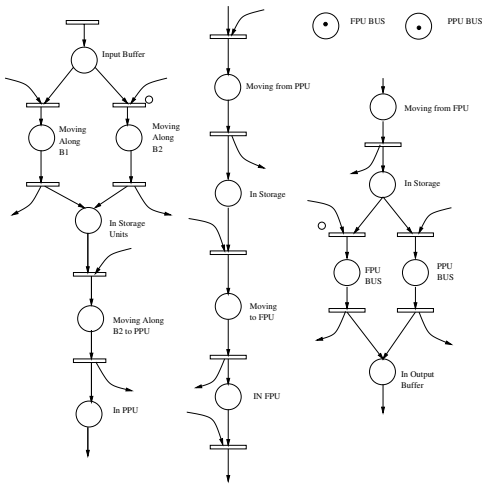

Figure 3: Petri Net representation of architecture

Figure 4 shows the system level schematic network modeling of this sequential Petri net system, which benefits the studying of system throughput, utilization, and analyzing the bottleneck in the system performance.

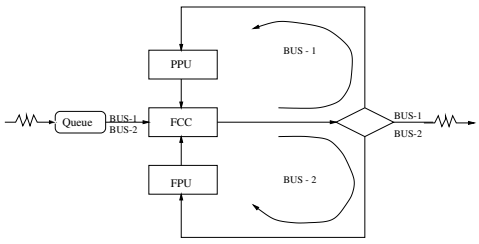

Figure 4: Schematic Network Model of Architecture

Based on the petri net representation in Figure 3, we build a performance simulation. There are several fluidic and pre/post processing units in the architecture. Each of them processes just one unit of incoming fluid sample. Storage buffers include many cells of the same volume. There is only one bus between the FPU and FCC (the primary FPU bus), and one between the PPU and FCC (the PPU bus). The interarrival time of fluidic samples satisfies a certain probabilistic distribution.

Figure 5 shows throughput simulation results. The horizontal axis denotes the sample input rate. The vertical axis represents what proportion of the inputs are accepted and processed by the system. The straight line shows an ideal case when availability of the system is guaranteed and all input samples are accepted and processed. The other curve is actual system performance. When input rate is low, throughput is nearly linear – the performance of the system approximates the ideal when input rate is low. At increased input rates, actual

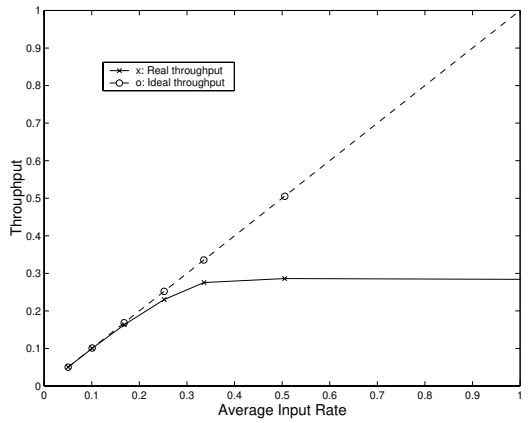

Figure 5: Actual vs. ideal system throughput

throughput drops below the ideal and quickly reaches saturation. At saturation, throughput remains constant regardless of input rate variation. Saturation is a common phenomenon in the study of system performance – it tells us that the resource is limited with respect to demand. The closer to the saturation point the system is operated, and the higher the saturation level of an architecture, the more efficiently we use available resources.

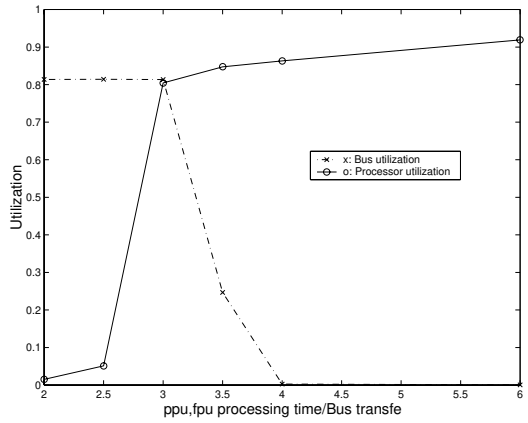

Figure 6: Bottle neck of system architecture

System architecture optimization is the key for higher system performance. Figure 6 shows the bottleneck of the system. The horizontal axis shows processing time normalized by bus transfer time. The vertical axis represents utilization. At low execution time ratios, transfering samples through the bus takes longer than processing samples. Therefore, the system bus is heavily utilized and processing units are under utilized (utilization below 0.1). There is a bottleneck at the point of communication. On the other hand, when bus transfer time is 1/4 that of processing time, processing becomes the bottleneck. More processing units would be needed to keep the system running at full speed. The ideal case

for this model is when bus transfer time is exactly 1/3 of processing time, where both communication resources and processing resources are well utilized.

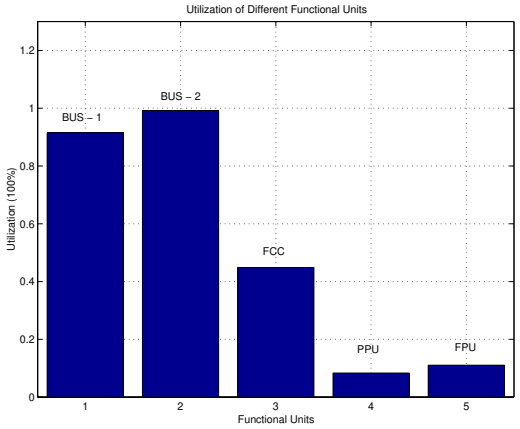

Figure 7: Utilization of different functional units

Furthermore, the hardware design of microsystems is also related to the control software design. Figure 7 shows the utilization of each functional unit with a certain fluidic processing route control schedule. Although either bus (primary FPU or PPU) may become the bottleneck to improving the system performance, the FCC is the center of MEF systems. Its architecture and control scheduling should be carefully designed, otherwise "Dead lock" due to cycle resource requirements between the related functional units may happen to make the system collapse.

5 CONCLUSIONS/FUTURE WORK

MEFs has advanced immensely in recent years. However, designs and devices are still largely application and function specific. Here, the design of a larger reconfigurable MEFS architecture has evolved from the demands that executing many application sets require. The system has logical functional units (FCC, FPU and PPU) and an interconnecting bus system that allows communication internally.

Performance simulation of such an architecture indicates that optimum performance is highly dependent upon the relationship between transit time and actual processing time. In addition, the distribution of process times associated within one functional unit (FPU or PPU) will play a role in optimizing bus and system usage. Furthermore, control scheduling is observed to also be a potential point for throughput bottlenecking or system dead-locking.

Armed with this new knowledge, further efforts plan on implementing physical models, exploring these relationships and seeking out other potential hazards and places for improvement. These discoveries are then fed back onto the design of the architecture to successively move closer to the design of a true laboratory on a chip.

6 ACKNOWLEDGMENT

This research effort is sponsored by the Defence Advanced Research Projects Agency (DARPA).

REFERENCES

[1] Emmanuel Delamarche et al. Patterned delivery of immunoglobulins to surfaces using microfluidic networks. *Science*, 276:779–781, May 1997.

[2] Emmanuel Delamarche et al. Microfluidic networks for chemical patterning of substrates: Design and application to bioassays. *Journal of the American Chemical Society*, 120:500–508, 1998.

[3] Daniel Figeys, Steven P. Gygi, Graham McKinnon, and Ruedi Aebersold. An integrated microfluidics-tandem mass spectrometry system for automated protein analysis. *Analytical Chemistry*, 70(18):3728–3734, September 1998.

[4] C. González, R. L. Smith, D. G. Howitt, and S. D. Collins. Microjoinery: Concept, definition, and application to microsystem development. *Sensors and Actuators A*, 66(1-3):315–332, 1998.

[5] Hal Jerman. Electrically-activated, normally-closed diaphragm valves. *Journal of Micromechanics and Microengineering*, 4:210–216, 1994.

[6] Carlos H. Mastrangelo, Mark A. Burns, and David T. Burke. Microfabricated devices for genetic diagnostics. *Proceedings of the IEEE*, 86(8):1769–1787, August 1998.

[7] W. Menz and A. Guber. Microstructure technologies and their potential in medical applications. *Minimally Invasive Neurosurgury*, 37:21–27, 1994.

[8] J. Pfahler, J. Harley, and H. Bau. Liquid transport in micron and submicron channels. *Sensors and Actuators A*, 22(1-3):431–434, 1990.

[9] Dieter Schmalzing et al. DNA typing in thirty seconds with a microfabricated device. *Proceedings of the National Academy of Science USA*, 94:10273–10278, September 1997.

[10] F. C. M. van de Pol, H. T. G. van Lintel, M. Elwenspoek, and J. H. J. Fluitman. A thermopneumatic micropump based on micro-engineering techniques. *Sensors and Actuators A*, 21(1-3):198–202, 1990.

[11] Patrick A. Walker, III, Michael D. Morris, Mark A. Burns, and Brian N. Johnson. Isotachophoretic separations on a microchip: Normal Raman spectroscopy detection. *Analytical Chemistry*, 70(18):3766–3769, September 1998.

[12] M. J. Zdeblick et al. Thermopneumatically actuated microvalves and integrated electro-fluidic circuits. In *Solid-State Sensors and Actuators Workshop*, 1994. Technical Digest.