AQUACORE: A GENERAL-PURPOSE ARCHITECTURE FOR PROGRAMMABLE MICROFLUIDICS

Ahmed M. Amin[1], Mithuna Thottethodi[1], T. N. Vijaykumar[1], Steven Wereley[2] and Stephen C. Jacobson[3]

[1]School of Electrical & Computer Engineering, Purdue University, [2]School of Mechanical Engineering, Purdue University ,[3]Department of Chemistry, Indiana University

Abstract

Interest in μTAS/LoC systems has grown considerably in recent years. However, the application-specific nature of current μTAS/LoC systems requires a new microfluidic chip to be designed for each type of assay, which incurs significant design effort, turn-around time, and cost, thereby reducing the productivity of μTAS users and slowing their widespread use. To that end, we propose a general-purpose, programmable LoC (PLoC), called AquaCore and demonstrate its use for two simple assays.

Keywords: General purpose chip, programmable microfluidics, AquaCore, AIS

1. INTRODUCTION

The AquaCore architecture provides a general-purpose design capable of executing a large class of assays, where the actual assay steps are described in a sequence of steps, called "fluidic instructions". In Table 1, we provide a comprehensive fluidic instruction set, called AquaCore Instruction Set (AIS), which is analogous to an assembly language used for programming computers. Compiler tool-chains can be developed to translate assays specified in easy-to-understand, high-level languages to AIS. Figure 1 illustrates the AquaCore architecture that implements AIS. The "wet" part of AquaCore consists of reservoirs, various fluidic functional units (FFUs) such as mixers and heaters, and a valve-and-channel fabric that can move fluid from any reservoir/FFU to any other FFU/reservoir by actuating the appropriate valves and pumps. The AIS instructions are interpreted by a "dry" microelectronic controller, which actuates the valves, pumps, and FFU controls (e.g., mixer operation and incubator temperatures) to achieve instruction execution.

In contrast to AquaCore's generality, previous work has explored programmability in a limited sense. Recently however, Urbanski et al. [1] replace these limited approaches with the pioneering idea of making LoCs fully programmable. In later work [2] they focus on a new programming language for microfluidics called BioStream, while we focus on the instruction set and architecture.

2. RESULTS AND DISCUSSION

Figure 2(a) and Figure 3(a) show two simple assays: DNA ligation [3] and RT-PCR [4], and Figure 2(b) and Figure 3(b) show their corresponding programs written in AIS. While computers are universal (i.e., they can perform any computation) due to Turing-Completeness of instruction sets and their implementations, fluidics lack such theoretical guarantees. Therefore, we must turn to empirical analysis of the AIS's coverage of real assays from different domains. We show in [5] that our architecture covers assays from genomics, biochemistry, chemical synthesis and immunoassays.

3. CONCLUSIONS

By providing a general-purpose programmable architecture and AIS, we enable μTAS developers to write their custom assays to run on this chip. We believe that the transition from LoCs to PLoCs is a paradigm shift in microfluidics, analogous to that in electronics

from calculators to programmable computers. This shift should enable the ubiquitous use of PLoCs and should allow μTAS users to focus on assay design rather than device design, thus boosting productivity.

Table 1. AquaCore Instruction Set (AIS)

Instruction	Description
move *id2, id1, <vol>*	Move the contents of *id1* to *id2*. *id1* & *id2* can be a reservoir, FFU or input/ output port. *vol* is an optional parameter to specify the relative volume to be moved. If no volume is specified, a default is used.
move-abs *id2*, *id1*, *vol*	Similar to move except *vol* is not optional and it specifies absolute volume in nl.
mix *id*, *time*	Mix the contents present in mixer *id* for a duration of *time* seconds.
separate.SIZE *id*, *time*	Separate the contents of the size-based separator *id* into two parts based on their size by passing the contents through an inbuilt filter for a duration of *time* seconds. The two output parts are stored internally in separate pools with unique ids that can be moved out later.
separate.CE *id*, *Esep*, *len*, *time*	Using capillary electrophoresis (CE), separate the contents of the separator *id* by waiting for a duration of *time* seconds. The contents pass through a CE column of length *len* under an electric field of *Esep*. If *len* is more than the hardware length, the fluid is automatically cycled for multiple iterations. Separation buffer is moved to the column's buffer port which has a unique id. For gel-based separation, the gel is preloaded into the column's gel port.
separate.AF *id*, *time*	Perform affinity-based separation in separator *id* for a duration of *time* seconds. The two outputs are held in separate pools with unique ids that can be moved out later. The FFU has a special internal pool (with a unique id) to hold the affinity-matrix which is moved there.
incubate *id*, *temp*, *time*	Heat the contents in the heater *id* to a temperature *temp*, incubate the contents at that temperature for a duration of *time* seconds.
input *id2*, *id1* **output** *id2*, *id1*	Input the fluid from the input-port *id1* into the reservoir *id2* Output the fluid stored in the reservoir *id1* to the output port *id2*
coated-plate-setup *id*	Prepare the reaction plate *id* that needs to be coated with specific chemicals/biomolecules. The coating is done by pressure-evaporating the coating compound onto the plate. The compound is dispensed from an input port dedicated to the plate.
sense.OD *id, senseval*	Sense the optical density of the fluid in the optical sensor *id* and copy the sensed value to the dry variable *senseval*. Instruction completion is detected by feedback from the sensor which is typically off-chip.
sense.FL *id, senseval[]*	Continuously sense the fluorescence in the contents of the CE column associated with the fluorescence sensor *id* and periodically copy the sensed value to the dry variable array *senseval*[].
concentrate.EV *id, temp, time*	Concentrate the mixture in heater *id* by evaporating the carrier fluid by holding the mixture at temperature *temp* for duration *time*. Other methods for concentrating (e.g., electrophoresis, dielectrophoresis, precipitation) may be adopted by extending AIS to add other variants of the concentrate instruction.
dispose *id*	Dispose of the contents of *id* which can be a reservoir or FFU.

REFERENCES

[1] J.P.Urbanski, W.Thies, C.Rhodes, S.Amarasinghe, T.Thorsen, *Lab on a Chip*, **6**(1):96–104, 2006.

[2] W. Thies, J. P. Urbanski, T. Thorsen, S. Amarasinghe, *DNA*, **12** (2006).

[3] J.W.Hong, V.Studer, G.Hang, W.F.Anderson, S.R.Quake, *Nature Biotechnology*, **22**(4):435-439, 2004.

[4] L.Warren, D. Bryder, I.L.Weissman, S.R.Quake, PNAS, **103**(47): 17807-17812, 2006.

[5] A.M.Amin, M.Thottethodi, T.N.Vijaykumar, S.Wereley, S.C.Jacobson, ISCA 34, 2006.

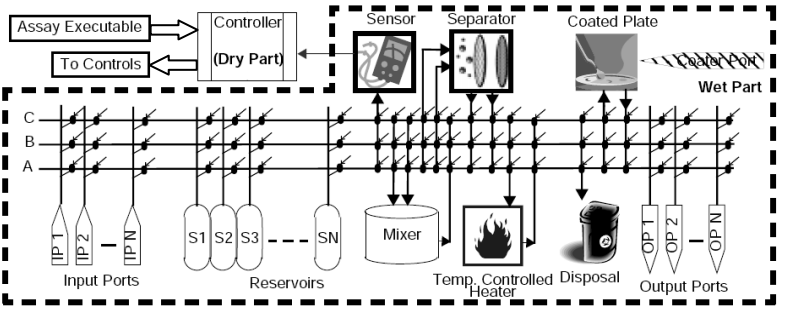

Figure 1. AquaCore Microarchitecture

(a)

1. Load cell suspension, dilution buffer and lysis buffer
2. Load beads into affinity column port
3. Load wash and elute buffers
4. Cell lysis: mix the dilution buffer, cell suspension and lysis buffer
5. Wash lysate
6. Recover purified DNA

(b)

```
DNA-extraction {
input s1, ip1          ;cell suspension
input s2, ip2          ;dilution buffer
input s3, ip3          ;lysis buffer
input s4, ip4          ;beads
input s5, ip5          ;wash buffer
input s6, ip6          ;elute buffer

move separator1.matrix, s4
move separator1.wash1, s5
move separator1.wash2, s6

move mixer1, s1
move mixer1, s2
move mixer1, s3
mix mixer1, 240

move separator1, mixer1
separate.AF separator1, 30
dispose separator1.flow-through
output op1, separator1
}
```

Figure 2. DNA Ligation
(a) Assay (b) AIS program

(a)

1. Load cells, buffer and sort cells
2. Perform reverse transcription
   a. mix cells and RT buffer
   b. heat to 55ºC for 15 min
   c. heat to 70ºC for 5 mins
3. Perform PCR
   a. heat to 95ºC for 3mins (warm up)
   b. do for 40 cycles:
      i. heat to 95ºC for 15s (denature)
      ii. heat to 60ºC for 60s (annealing)

(b)

```
RT-PCR {
input s1, ip1     ;cell suspension
input s2, ip2     ;RT-PCR buffer
input s3, ip3     ;reverse transcriptase
input s4, ip4     ;DNA polymerase
input s5, ip5     ;gene specific primers & probes

move separator1.buf, s2
move separator1, s1
separate.AF separator1, 30
move mixer1, s3
move mixer1, s4
move mixer1, s5
move mixer1, separator1
mix mixer1, 30
move heater1, mixer1
incubate heater1, 55, 900
incubate heater1, 70, 300
incubate 95, 180
dry-mov r1, 40
dry-label loop:
  incubate 90, 15
  incubate 60, 60
dry-dec r1
dry-bgt loop
}
```

Figure 3. RT-PCR
(a) Assay (b) AIS program