UNIVERSITY OF CALIFORNIA
RIVERSIDE

Reducing Control Hardware of Soft Robotics With Pneumatic Logic

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Bioengineering

by

Shane Sang Hoang

December 2022

Dissertation Committee:

Dr. William H. Grover, Chairperson
Dr. Philip Brisk
Dr. Elena Kokkoni
Dr. Konstantinos Karydis

The Dissertation of Shane Sang Hoang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

Thanks to Dr. William Grover for advising me through this academic journey and exuding an infectious enthusiasm and excitement for my research that always helped to lift my spirits whenever I felt lost. I have great fondness for the lab space you've fostered over the past decade and cherished every opportunity I had to come in and learn something new. Also a special thanks to my graduate advisor Dr. Hyle Park for assisting in my transition from UC Davis to UC Riverside and providing me with funding opportunities to kickstart my doctoral studies. Graduate school started out on a rough note but the assistance I received from Will and Hyle throughout my time at UCR was critical in allowing me to navigate through any roadblocks I came across.

Thank you to the members of my oral qualifying exam and thesis committee, Professors Philip Brisk, Elena Kokkoni, and Konstantinos Karydis, for collaborating with me on my research and providing feedback on drafts of my research papers.

I owe a great deal of my accomplishments to Heran Bhakta for spotting qualities in me worth taking a chance on during a time when an abrupt end to my graduate studies seemed imminent. Thank you for providing the push I needed to be direct in resolving matters at hand rather than taking roundabout approaches; without your help, I likely wouldn't have made it this far through the doctoral program. I appreciate all the time you spent teaching me about microfluidics and life in general.

Thank you to my fellow grad students, Sam and Ray, and undergrads, Mabel, Miles, Zinal, and Huy, in lab for helping to liven up our lab group following the pandemic. With the campuswide shutdown of labs and limited in-person interactions even after re-

strictions were lifted, having labmates to talk to who were active in pursuing research was a breath of fresh air and made my last year at UCR an enjoyable one. I greatly enjoyed all of our lab outings and hope they become a staple in the coming years of our lab.

Thanks Dr. Scott Simon for advising me during my undergraduate research at UC Davis and especially Vasilios Morikis for guiding me down the path of research in Simon Lab. V, watching how you genuinely enjoyed your research and had fun every day in lab inspired me to pursue graduate school to find research that I could also be as passionate about. I hope I've become even a fraction of the amazing scientist that you are.

Graduate school ended up being the best years of life thanks to the countless memories I made with great friends along the way. Thanks to Sid, Wayne, Martina, George, Jack, Thompson, Heran, Brandon, Lauren, Kofi, Lisa, Sam, and many others for creating a space where I felt I could truly be myself without any worry. Our Friday game nights were always the highlight of my week and made all these years of graduate school pass by in the blink of an eye. I'm fortunate to have been able to meet such great-minded individuals and I will always treasure our friendships.

Finally, thank you to my parents and siblings for preparing me to take on the world as I grew up in our home in Stockton. Whether I was going to school in Stockton, Davis, or Riverside, the one constant was the family I could always return to. Being unable to get any work done whenever I come home is a testament to just how much fun I have spending time with you all. Knowing that I will always have the emotional support of each and every one of you makes navigating through the twists and turns of life much more manageable.

To my parents

Nam Hoang and Vuong Nguyen

for all their sacrifices and loving support

ABSTRACT OF THE DISSERTATION

Reducing Control Hardware of Soft Robotics With Pneumatic Logic

by

Shane Sang Hoang

Doctor of Philosophy, Graduate Program in Bioengineering
University of California, Riverside, December 2022
Dr. William H. Grover, Chairperson

During an age of technological innovation, automation has become more and more prevalent in various industries, leading to the development of robots to assist in performing tasks in a precise manner, free of human variation. Over the past decade, soft robotics has emerged as a field that aims to "soften" these robots by replacing the circuitry and rigid, metallic bodies with flexible materials in order to make interactions with humans safer, all while maintaining the functionality of the traditional robots. These soft robots are powered through pneumatics rather than electricity, with the manipulation of the empty space of air within the hollow, plastic constructs allowing for actuation of the robot. Stemming from a biomimetic approach, soft robots have proven capable of demonstrating semblances of motor skills and swimming gaits seen in other living organisms; however, the precise timing of movements still requires the reliance on electronics. As such, this study aims to develop approaches to minimizing the footprint of the electromechanical control hardware by employing pneumatic logic.

The concept of pneumatic logic entails the use of air to power logic circuits, which

can be designed by arraying together normally-closed valves, formed using polydimethylsiloxane (PDMS) as a flexible membrane, in a microfluidic chip. This results in the development of pneumatic logic gates that act similarly to their electronic counterparts (e.g. AND, OR, NOT), with the key difference being that the pneumatic analog can be used to impede the flow of air rather than electricity, making them suitable for controlling soft robots. Powered by vacuum pressure in particular, this use of monolithic membrane valves in conjunction with pneumatic logic minimizes the electronic control hardware required to operate multiple soft robots independently, with error-checking capabilities present as well. This design also enables a mode of continuous soft robotic actuation without the need for any controllers, requiring only a source of vacuum pressure. Able to operate with low pressures and at the microfluidic scale, pneumatic logic presents itself as an attractive option for controlling soft robots in an efficient manner.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Microfluidics describes the principle of controlling a small volume of fluid, commonly using channels with widths in the range of 10^{-6} – 10^{-9} L. The use of such a minute quantity has allowed it to become a powerful tool in the realm of bioanalysis, as the small volume enables for analyses to be carried out at a lower cost (less reagents being used) and at a much faster rate. As such, microfluidic circuits have been developed to carry out multiple lab operations in parallel, leading to the coined term lab-on-a-chip. For example, by designing a chip where channels intersect, junction points become areas where reagents can be mixed. Adjusting the resistances of one of these channels (e.g. changing the length or width) can create a time-delay for a secondary reagent. Having channels continuously intersect and branch out can lead to a concentration gradient. All in all, microfluidic chips provide a means of making many of these bioprocesses more efficient while also introducing on-chip control mechanisms for governing fluid flow.

These microfluidic layouts can be placed into the category of passive microfluidics,

where fluid flow is governed by capillary flow. Chips that use this type of microfluidics rely on passive, or inherent, attributes of the design to characterize how fluids will be driven. Alternatively, microfluidic chips can use active components that can be used as gates to control whether fluids can pass through at a given time. To create these, researchers took inspiration from electronics, where transistors are used as gates in microprocessors to control the flow of electricity, leading to the implementation of microfluidic valves and pumps. These valves and pumps originated in the form of Quake valves, which were designed by Stephen Quake and consisted of three components (a fluidic layer, a control layer, and a flexible membrane) [87]. After forming the channels (through soft lithography) on two separate substrates, these layers are bonded such that the channels are facing each other and intersect, with the flexible membrane in-between. When there is no pressure supplied to the control layer, fluids can freely flow from the inlet of the fluidic channel to the outlet. However, providing positive pressure through the control channel will cause it to stretch and push on the flexible membrane, thereby moving it into the fluidic channel and impeding flow. Further application of this valve concept can be extended to making a peristaltic-like pump, where having three valves in series and actuating them in turn can push fluid through the channel [25]. This pioneered the use of a flexible membrane as a means of controlling the flow of fluids within microfluidic chips, leading to the development of active microfluidics.

While Quake valves are normally open at rest, valves have also been designed to operate from a normally-closed state, as demonstrated by William Grover [25]. The key difference in the latter's design is the use of vacuum, rather than positive pressure, to operate the flexible membrane, namely polydimethylsiloxane (PDMS), as a valve. As shown in Fig

1.1, the layout of these valves is composed of a fluidic channel on one side of a substrate and a control channel leading to a displacement chamber on the other, with these features being fabricated through photolithography and wet chemical etching on glass substrates [25]. The chamber acts as a space beneath the valve where the PDMS can be pulled into. Inlet and outlet reservoirs are then formed by drilling through the glass at desired locations to provide openings for interfacing with the etched channels. Following these steps, the two glass substrates are bonded together with a thin sheet of PDMS between the layers.

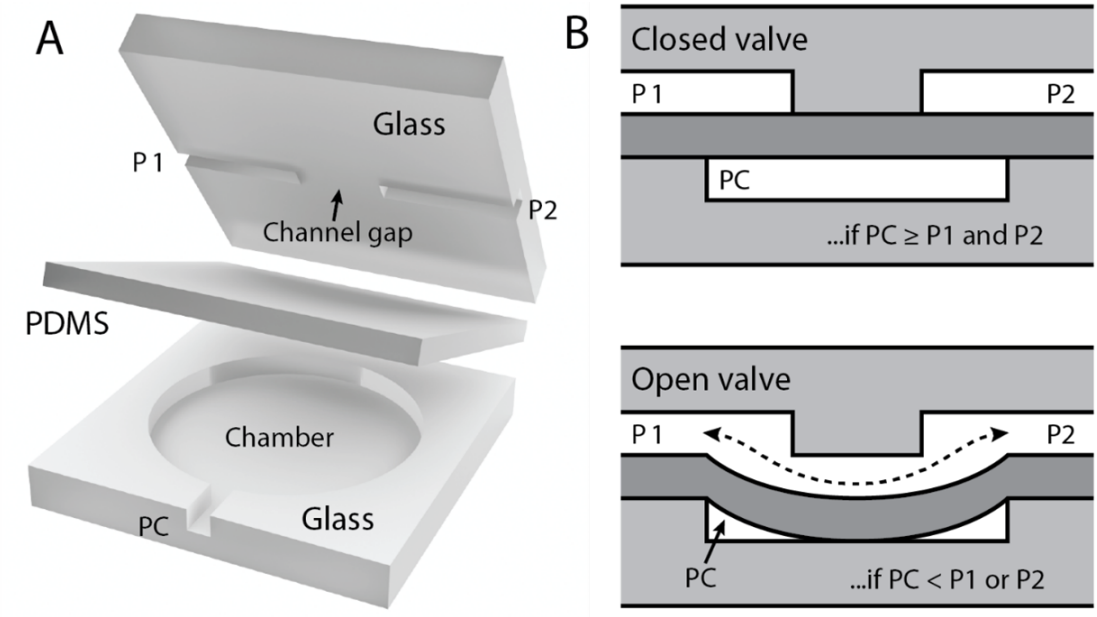

Figure 1.1: Exploded (A) and cross-sectional (B) views of monolithic membrane valves. Differences in pressure between the control channel and inlet/outlet cause the PDMS membrane to be pulled down.

Similarly to Quake valves, the fluidic channel carries fluids from the inlet to the outlet; however, being that these are normally-closed valves, the channel is left disconnected with unetched glass filling the gap between the ends, thus preventing any fluids from flow-

ing through at rest. To open the valve, vacuum pressure can be supplied to the control channel, causing the PDMS membrane to be pulled into the displacement chamber, thereby connecting the ends of the fluidic channel and allowing fluids to pass over the valve. While seemingly just providing an alternative means of controlling a valve (positive vs. vacuum pressure), normally-closed valves formed from PDMS membranes have provided a strong foundation for fabricating more complex microfluidic chips.

As described previously, passive and active microfluidics differ in how fluid flow is governed. While flow behavior in passive microfluidics is dependent on capillary flow, active microfluidics takes advantage of active components, such as normally-closed microvalves that form micropumps to manipulate fluid flow. Although most people associate the term "fluid" with just liquids, it can also refer to gases. As such, gases, such as the ambient air, can be carried through microfluidic chips as well using valves and pumps. The advantages of controlling air over liquids stem from the former's compressibility and low viscosity at normal temperatures, which allow it to be easily manipulated with membrane valves [19, 2]. By using active components to control air, pumps can continuously drive the fluid while the valves determine the direction that it will take. Extending this notion further, multiple valves can be arrayed together in a circuit to form logic gates, similar to those seen in electronics, thus introducing the concept of pneumatic logic. Similarly to how its electronic counterpart controls the flow of electrons in a circuit, pneumatic logic gates act to control the flow of fluids, namely air in this case. Using normally closed-valves, pneumatic equivalents of previously established logic gates, such as AND, OR, NOT, NAND, and NOR, can be designed as shown in Fig 1.2. These logic gates illustrate scenarios where

a source of vacuum pressure at the inlet is used to pull air from the outlet. Within each gate, truth tables govern how fluids flow depending on the state of the valves. Given the binary nature of logic gates and that the normally-closed valves are pneumatically-powered, valves at rest are represented by a value of FALSE or 0 (atmospheric pressure is supplied to the displacement chamber) while actuated valves have a value of TRUE or 1, indicating that vacuum pressure is being supplied to the displacement chamber to pull the membrane down. Lastly, the overall output of the valve is measured at the output where a value of 1 represents vacuum pressure and a value of 0 represents atmospheric pressure. Looking at the AND and NOT gates, the two most commonly known logic gates in circuits, it can be seen that these gates are made up of two valves in series and two valves in parallel, respectively. As such, air is only able to pass through the AND gate when both Valves A and B are pulled down (or have values of 1) and through the OR gate when either of the valves are opened. Vents to atmospheric pressure, represented by darkened circles in Fig 1.3, can also be used to direct fluid flow due to fluids taking the path of least resistance. As exemplified by the NOT (or inverter) gate, air typically flows from the outlet to the inlet unperturbed, but once Valve A is opened, a path of lower resistance is formed between the inlet and the vent, causing the air to be drawn in from the opening rather than the outlet. These logic gates demonstrate how varying arrangements of valves can be used to add additional modes of control for manipulating fluid flow. Furthermore, this opens up opportunities for automation where microfluidic chips are able to operate continuously through feedback loops.

Being able to execute certain tasks within microfluidic chips in an autonomous

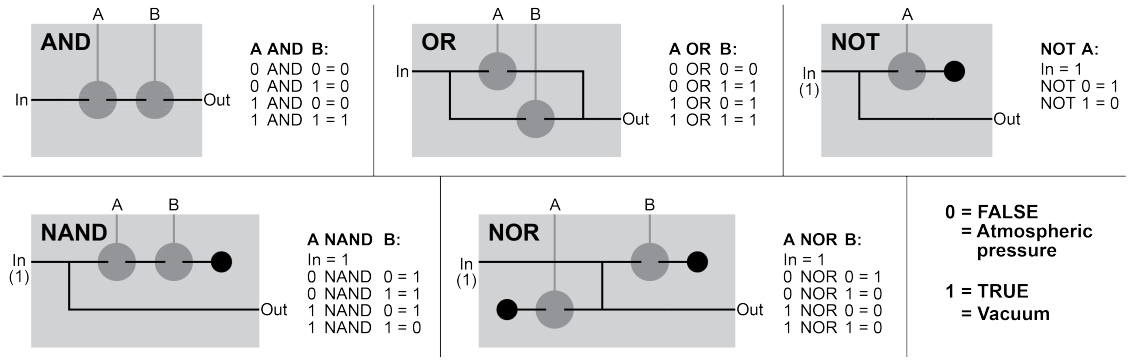

Figure 1.2: Pneumatic logic gates with associated truth tables, where vacuum pressure is being used as a source for the inlet and operating valves A and B.

manner has long been a major roadblock in the development of complex systems on these devices [19]. While the use of external hardware and electronics can simplify these devices by providing the user with a wide range of control over the processes occurring within the chip, these additions make the system bulkier, more expensive, and less accessible to underdeveloped nations. To address this issue, researchers have started looking into ways of automating the control of microfluidic systems through on-chip methods, with one prominent solution being the adaptation of a ring oscillator [19]. By connecting an odd number of inverter or NOT gates together in series, oscillation will occur due to the inversion of the initial signal. As shown by the electronic schematic of a ring oscillator in Figure 3, sending a signal through three inverters causes the signal to become the inverse of its initial state, which is then fed back as an input through a feedback loop. This new input signal will then go through the NOT gates and become inverted, fed back into the loop, and cause oscillation as the process is continually repeated.

The equivalent of the ring oscillator in microfluidic chips would be achieved by

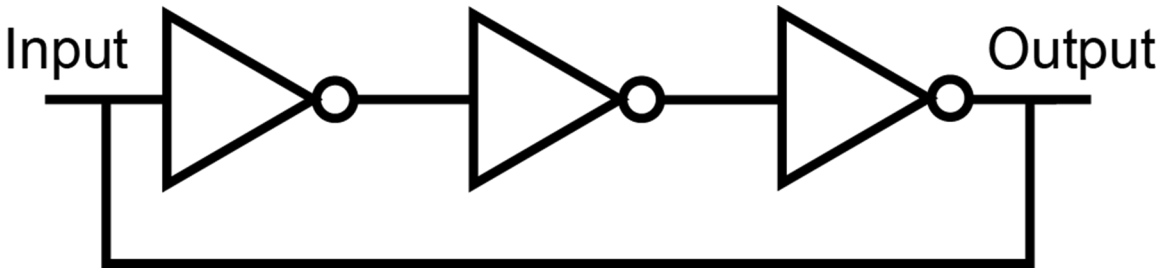

Figure 1.3: An electronic logic diagram for a ring oscillator, where three inverter or NOT gates are placed in series. The output signal of this circuit becomes the inverse of the input signal and creates oscillation as it is used as an input signal for the next cycle through a feedback loop.

connecting three inverter valves in series. With each of the valves being dependent on the previous valve, a vacuum source can be used to cause these valves to oscillate continuously, with a schematic of an oscillator valve design shown in Fig 1.4. In this diagram of a microfluidic chip with normally-closed valves, "V" is used to represent a vacuum source while "A" represents an opening to atmospheric pressure. The green channels depict where the initial vacuum is drawing air while the red channels show where there is atmospheric pressure. Transparency is used to show that the control channels are present on a different layer than the channels directly connected to the vacuum and atmospheric pressure sources. These two layers are separated by a PDMS membrane with openings in the membrane shown as ports. When the vacuum source is turned on in Fig 1.4A, it will pull in air from each of the three channels, but since Valves 1,2, and 3 are normally closed, air will be drawn from the side channels present in each respective channel. Because each side channel is connected to the control channel of the next valve through the respective ports, each valve set acts as an inverter gate and causes the vacuum source to draw air from each valve, thereby depressurizing and opening them in series, as shown in Fig 1.4B. Once the valves are pulled down by the vacuum, the path to the atmosphere will be open, which will cause the vacuum

to draw in atmospheric pressure that will flow through the chip and close all the valves (Fig 1.4C). This in turn resets the chip in Fig 1.4D to its previous configuration where the valves are closed and causes the vacuum source to open them again, resulting in an unsteady state where the circuit acts as a feedback loop for itself. Consequently, the combination of three inverter gates (or any odd number of inverters) in series creates a chain of oscillating valves, which researchers have been able to use to automate pumps within microfluidic chips as well as make clocks for timing references [19]. This form of microfluidic technology opens many avenues for microfluidic systems by introducing analog modes of control rather than discrete digital inputs.

In addition to reducing the complexity of control mechanisms used to manipulate fluids within active microfluidics, pneumatic logic also offers sophisticated on-chip fluid control. While passive microfluidics chips manipulate fluids through passive means, such as gravity, capillary flow, and diffusion, active microfluidic chips can make use of logic gates to allow fluids to move in certain paths once specific conditions are met at given times. This also allows both halves of the valve, namely the fluidic and control channels, to be supplied with the same fluidic source, as opposed to having liquids travel through the fluidic channel and the valve be operated with air. In the case of the latter, there were only two states the valve could be in: open or closed depending on whether it was a normally-closed or normally-open valve and whether pressure (positive or vacuum) was supplied to the control channel. While using air-powered microfluidic pumps and valves to control the flow air does not have direct applications to common lab operations (due to the lack of liquids/reagents), it does introduce new modes of control and logic into microfluidics, akin to those seen in

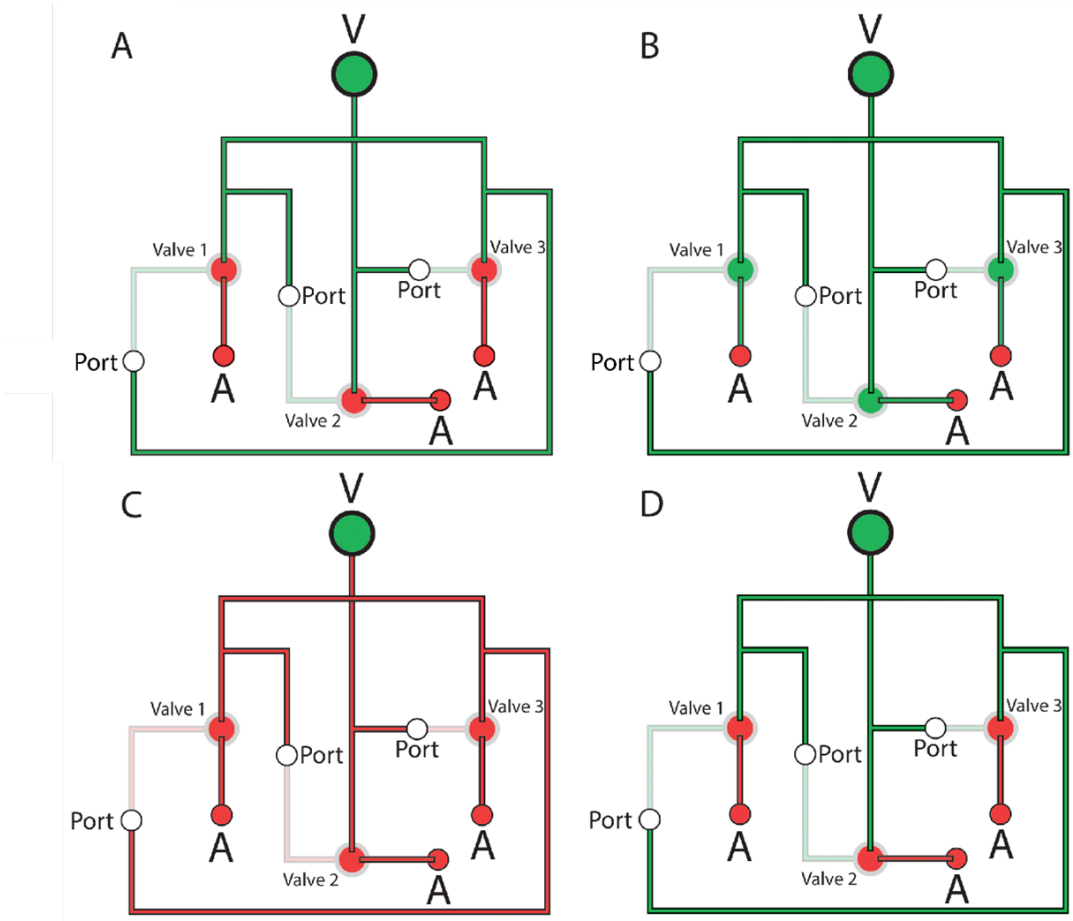

Figure 1.4: A microfluidic chip with a series of inverter or NOT gates. The vacuum source is depicted by "V" while sources of atmospheric pressure are depicted by "A". Green channels are used to represent the path where air is drawn to the vacuum source while red channels show where atmospheric pressure is being drawn in. Transparency is used to show that the control channels are present on a different layer than the channels directly connected to the vacuum and atmospheric pressure sources, with ports representing an opening between the two layers. (A) A vacuum source draws air throughout the chip. (B) The vacuum pulls air from the control channels and opens the valves. (C) Air is drawn in from the atmosphere which rushes through the chip and closes the valves. (D) The closing of valves resets the chip and causes continuous oscillation due to the unsteady state of the three inverter gates in series. A detailed explanation of the operation of the chip is provided in the text.

electronics. One specific case can be seen when the vacuum pressure supplied to both the fluidic and control channels balance out.

As mentioned earlier, according to pneumatic logic, when the displacement chambers below normally-closed valves are supplied with vacuum pressure, the PDMS membrane gets pulled down, opening an airway between the inlet and outlet and allowing flow to ensue in one direction. This truly only applies in scenarios where the vacuum pressure below the membrane valve exceeds the vacuum pressure in the inlet and outlet. When the vacuum pressure in the inlet/outlet exceeds the vacuum pressure supplied to the displacement chamber, the membrane valve remains shut. What would happen in the case where the sources of vacuum on each side of the PDMS membrane equate? While it seems logical to believe that the membrane will also stay shut since the pressures are equal, these conditions actually cause the membrane to open until the pressures in the inlet, outlet, and control channel equate, at which point the valve will close on its own [24]. The reason being is that in using vacuum pressure to open the valve as well as draw air from the outlet to the inlet, there is actually an imbalance of pressure since the outlet is at atmospheric pressure. It's this pressure difference that allows air to flow from higher (atmospheric) to lower (vacuum) pressure in the first place. If the outlet in this thought experiment is directly connected to the atmosphere, then air will continuously flow from the outlet to the inlet due to this constant pressure gradient. However, if the outlet was instead connected to a finite space, the pressure imbalance would only remain intact for as long as the pressure at the outlet remains greater than that at the inlet. As the air in the outlet is slowly pulled towards the inlet, the pressure on this side will start to decrease until ultimately reaching

vacuum pressure, causing the pressures in all three channels to equate and the membrane valve to close [24]. Such valves have been coined as 'latching valves' and further increase the complexity of control within microfluidic circuits by introducing valves that can close independently under given conditions, thus allowing for multiple operations to occur on the chip simultaneously. Furthermore, the monolithic nature of these membrane valves allows for large-scale fabrication of these microfluidic chips in one operation, requiring only sources of vacuum to impart the latching behavior.

Electromechanical systems have played a critical role in the development of the world. Due to the contributions of inventors and engineers over the past centuries, countless devices and equipment have been designed and refined to produce means of automating important processes that have proven key in improving the overall quality of life within society. New instruments are being developed to allow for precise monitoring of patients in hospitals as well as assisting in the research for cures to prevalent diseases. Public transportation systems are always improving to allow passengers to safely connect with others over great distances in as an efficient manner as possible. Advancing technology is increasingly being integrated into household applications to ease burdens on families through smart controls and enhancements in security surveillance. As society continues to develop, automation is becoming more prevalent in all facets of everyday life. Consequently, the emergence of robots has been developing hand-in-hand with advances in automation.

Representing the pinnacle of automation, robots are becoming more prevalent as they integrate into society. Ranging from house cleaning to assisting in construction projects to mapping out extraterrestrial terrains, autonomous robots can be found in many

sectors of modern society. Designed to perform tasks as well as, if not better than, humans, robots are able to execute tasks with unparalleled precision and accuracy due to being unaffected by the variability of human motor controls. This has allowed robots to have an increasing presence alongside members of society. While the automated operations carried out by robots provide many benefits to human workers, this technological development is met with dangers as well. Specifically, most autonomous robots are designed to have tough, metal exteriors to make them more durable and have longer lifespans; however, their bulky, rigid bodies also pose health risks to the humans they work alongside in the case of unintended collisions. Furthermore, the electronic control hardware for these robots presents the possibility of sparks and explosions in hazardous conditions. In addition to these dangers, these features also make these rigid robots more expensive to maintain. These drawbacks have led to the development of soft robots, which are designed to have rubbery bodies for safer interactions with humans, to allow them to perform more delicate tasks, and to reduce fabrication and maintenance costs. However, many of the soft robots being developed still rely on external electronics to operate, presenting similar hazards as traditional hard robots and leaving them susceptible to malfunctioning in poor weather conditions. As such, this work will focus on introducing strategies for eliminating the electromechanical control hardware currently deployed in soft robotics, specifically through the use of pneumatic logic.

Chapter 2

A pneumatic random-access memory for controlling soft robots

Reprinted with permission from Hoang S, Karydis K, Brisk P, Grover WH (2021) A pneumatic random-access memory for controlling soft robots. PLoS ONE 16(7): e0254524. https://doi.org/10.1371/journal.pone.0254524

2.1 Abstract

Pneumatically-actuated soft robots have advantages over traditional rigid robots in many applications. In particular, their flexible bodies and gentle air-powered movements make them more suitable for use around humans and other objects that could be injured or damaged by traditional robots. However, existing systems for controlling soft robots currently require dedicated electromechanical hardware (usually solenoid valves) to maintain the actuation state (expanded or contracted) of each independent actuator. When combined

with power, computation, and sensing components, this control hardware adds considerable cost, size, and power demands to the robot, thereby limiting the feasibility of soft robots in many important application areas. In this work, we introduce a pneumatic memory that uses air (not electricity) to set and maintain the states of large numbers of soft robotic actuators without dedicated electromechanical hardware. These pneumatic logic circuits use normally-closed microfluidic valves as transistor-like elements; this enables our circuits to support more complex computational functions than those built from normally-open valves. We demonstrate an eight-bit nonvolatile random-access pneumatic memory (RAM) that can maintain the states of multiple actuators, control both individual actuators and multiple actuators simultaneously using a pneumatic version of time division multiplexing (TDM), and set actuators to any intermediate position using a pneumatic version of analog-to-digital conversion. We perform proof-of-concept experimental testing of our pneumatic RAM by using it to control soft robotic hands playing individual notes, chords, and songs on a piano keyboard. By dramatically reducing the amount of hardware required to control multiple independent actuators in pneumatic soft robots, our pneumatic RAM can accelerate the spread of soft robotic technologies to a wide range of important application areas.

2.2 Introduction

Pneumatically-actuated soft robots demonstrate certain advantages over rigid robots in many applications. For example, their soft rubbery bodies are suitable for use as robotic grippers for lifting delicate objects [20, 85, 75]. Their ability to yield increases their safety in close proximity to humans [92]. Soft robots are also suitable for use *in contact* with

humans, as wearable exoskeletons for assisting laborers, warfighters, the elderly, or patients with musculoskeletal or neurological conditions [14, 42]. Additionally, soft robots resemble living organisms more closely, which makes them suitable for use in biomimetics [86, 53, 83, 45, 37, 15, 47].

However, existing systems for *controlling* soft robots have significant disadvantages. Many soft robots move by using air under pressure or vacuum to expand or contract flexible actuators inside the robot. The flow of air to each independent actuator is usually controlled by a dedicated electromechanical solenoid valve, which in turn is controlled by an electronic or electromechanical relay, which in turn is controlled by an electronic microcontroller or computer. Ironically, all of this electronic and electromechanical hardware exists to control a robot that is, after all, fundamentally *pneumatic*, not electrical. This mishmash of two very dissimilar domains—pneumatics and electronics—makes current soft robotic systems unnecessarily complex, expensive, bulky, and power-hungry. And while some research has blurred the line between these dissimilar domains using *e.g.* electrically conductive and insulating fluids [21], these approaches still require electronic components and limit the use of these robots in potentially hazardous environments where these components could spark and cause a fire or explosion. Finally, electronic components also hinder the use of soft robots in wearable exoskeletons, therapeutic devices, and other applications in close proximity to humans where lugging around heavy batteries, valves, computers, and other electronics is impractical.

To solve this problem, researchers have turned to an idea that predates electronic computers: *pneumatic logic* [30]. In pneumatic logic, air (not electricity) flows through

circuits of tubes or channels, and air pressure is used to represent a logical state (on or off, TRUE or FALSE, etc.). In the decades before electronic logic became ubiquitous, principles of pneumatic logic provided advanced levels of control in a variety of products, including climate control systems (which used all-pneumatic thermostats, bellows, valves, and other components to control temperature and humidity throughout a building [1]) and player pianos (which used air to read punched-paper "programs" and to control nearly 100 independent keys in the early 1900s [62]). Since pneumatic logic is powerful enough to control buildings and pianos, it should also be capable of controlling soft robots.

However, the existing implementations of pneumatic logic for soft robot control have limitations that complicate their use with robots that contain many independent actuators. For example, the pneumatic logic gates and embedded actuators demonstrated by Preston *et al.* [69, 70] are fabricated one-by-one and connected together manually using tubing; this complicates their large-scale use in complex multi-actuator robots which could require tens or hundreds of logic gates. Other designs link pneumatic actuators together mechanically into pneumatic networks that demonstrate feedback-based oscillations [22, 94]; these are very useful for controlling repetitive operations (like walking gaits in legged robots) but are less suitable for individual actuator control and still require manual fabrication and assembly. Researchers have improved the manufacturability of pneumatic logic circuits by using microfabrication to create pneumatic logic circuits [7], but since these logic circuits use normally-open microfluidic valves [87] as transistor-like elements, they are limited to simpler logic circuits like demultiplexers which can only control one actuator at a time. This is because normally-open valves require a constant applied pressure to seal closed; when

this pressure is removed, the valves automatically reopen and vent any trapped pressures. With no way to store a pressure differential inside the device, pneumatic logic circuits built with normally-open valves have no "memory" and cannot maintain the state of one soft robotic actuator while setting another. Finally, while valves with bistable buckling silicone membranes have recently been used as single-bit memories in soft robots [58, 17], the large size and manual fabrication and interconnection of these elements again complicates their large-scale use in complex pneumatic logic circuits.

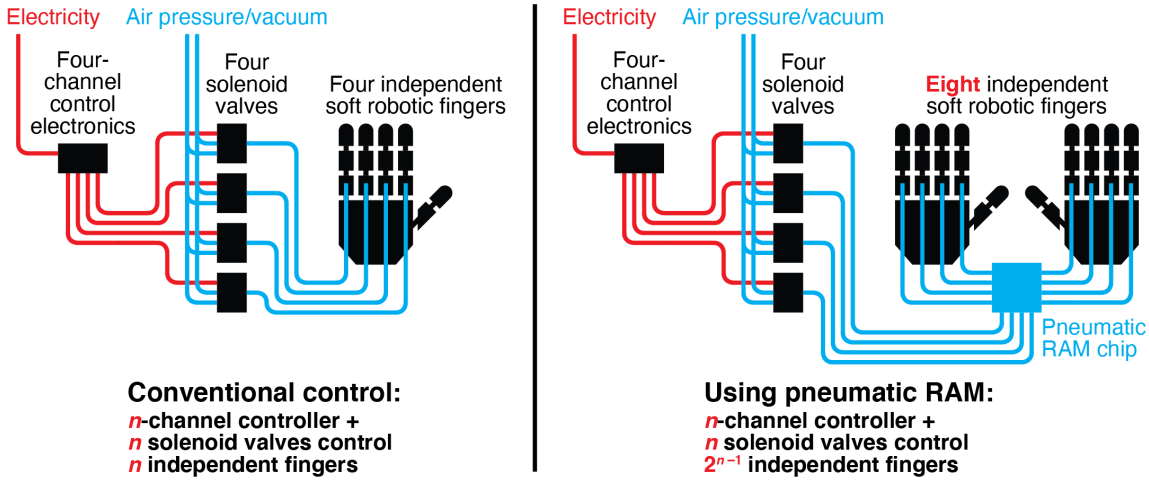

Figure 2.1: Conventional methods for controlling pneumatic soft robots require a dedicated electronic control line and solenoid valve for each independent actuator (left). Using our pneumatic random access memory (RAM), the same amount of electromechanical control hardware can operate many more actuators while still providing independent control of each actuator (right).

To address these limitations, we developed pneumatic logic circuits *with memory* that can be used to control large numbers of independent soft robotic actuators (Fig 3.1). We accomplished this by using *normally-closed* microfluidic valves [25] in our pneumatic logic circuits. Since these valves remain sealed against a pressure differential even when disconnected from a pneumatic control line, they can be used to create trapped pressure

differentials that function as pneumatic memories and maintain the states of large numbers of soft robotic actuators. And since these valves can be easily fabricated in dense arrays, they support complex circuits that perform advanced operations like time division multiplexing and pneumatic analog-to-digital conversion. These pneumatic logic circuits can significantly reduce the amount of expensive, bulky, and power-consuming electromechanical hardware required to control a pneumatic system.

2.3 Results

2.3.1 Adapting normally-closed microfluidic valves for controlling soft robots

Our pneumatic memory circuits use a modified form of monolithic membrane valves. These valves were originally developed for controlling fluid flow in microfluidic chips [25] and later used as transistor-like elements in pneumatic logic circuits for controlling microfluidic chips [34, 24, 19, 33, 36, 35, 39, 46, 59, 18]. Though perhaps not as well known as the soft lithography microfluidic valves developed by Stephen Quake [87], monolithic membrane valves have several traits that make them particularly useful in pneumatic logic applications. The two valving technologies are compared in Table 2.1.

Conventional monolithic membrane valves (Fig 2.2A) consist of an Input channel and an Output channel in one layer, a Control channel and chamber in a second layer, and a featureless polydimethylsiloxane (PDMS) silicone rubber membrane sandwiched between the two layers [25]. The valve is normally closed, meaning that when the Control chamber

Table 2.1: Comparisons between soft lithography valves (used in most previous work on microfabricated pneumatic logic control of soft robots) and monolithic membrane valves (adapted for use in this work).

	Soft lithography valves [87]	**Monolithic membrane valves** [25]
Fabrication:	Channels are cast in silicone rubber by pouring liquid silicone over a positive mold made using photolithography. After curing, the solid silicone channel layers are peeled off the mold. Two or more silicone layers are bonded together; valves are formed wherever two channels cross in different silicone layers.	Channels are etched in rigid glass using photolithography or engraved in rigid plastic using CNC milling. A featureless solid silicone rubber sheet is bonded between two rigid channel layers. Valves are formed wherever an etched/engraved chamber is located across the silicone sheet from a gap in a channel.
Operation:	A high pressure expands one channel into a second channel, pinching the second channel closed and closing the valve.	A low pressure in a chamber pulls the silicone sheet away from the gap in the channel, allowing flow across the gap and opening the valve.
Powered by:	Pressure	Vacuum
At-rest state:	Normally open	Normally closed
Suitability for use in pneumatic memory for controlling large numbers of independent soft robotic actuators:	Limited because normally open valves lose control of the contents of the valved channel when disconnected from a pressure source.	Favorable because normally closed valves maintain control of the contents of the valved channel when disconnected from a vacuum source. This enables trapped pressure differences that can serve as "memory" even when the valves are disconnected from power.

is at atmospheric pressure, the PDMS membrane seals against the gap between the Input and Output channels and blocks flow between them. When a vacuum is applied to the Control channel, the PDMS membrane is pulled into the Control chamber and away from the Input and Output channels, thereby opening a path for flow between the Input and Output channels.

Due in large part to their normally-closed nature, monolithic membrane valves have

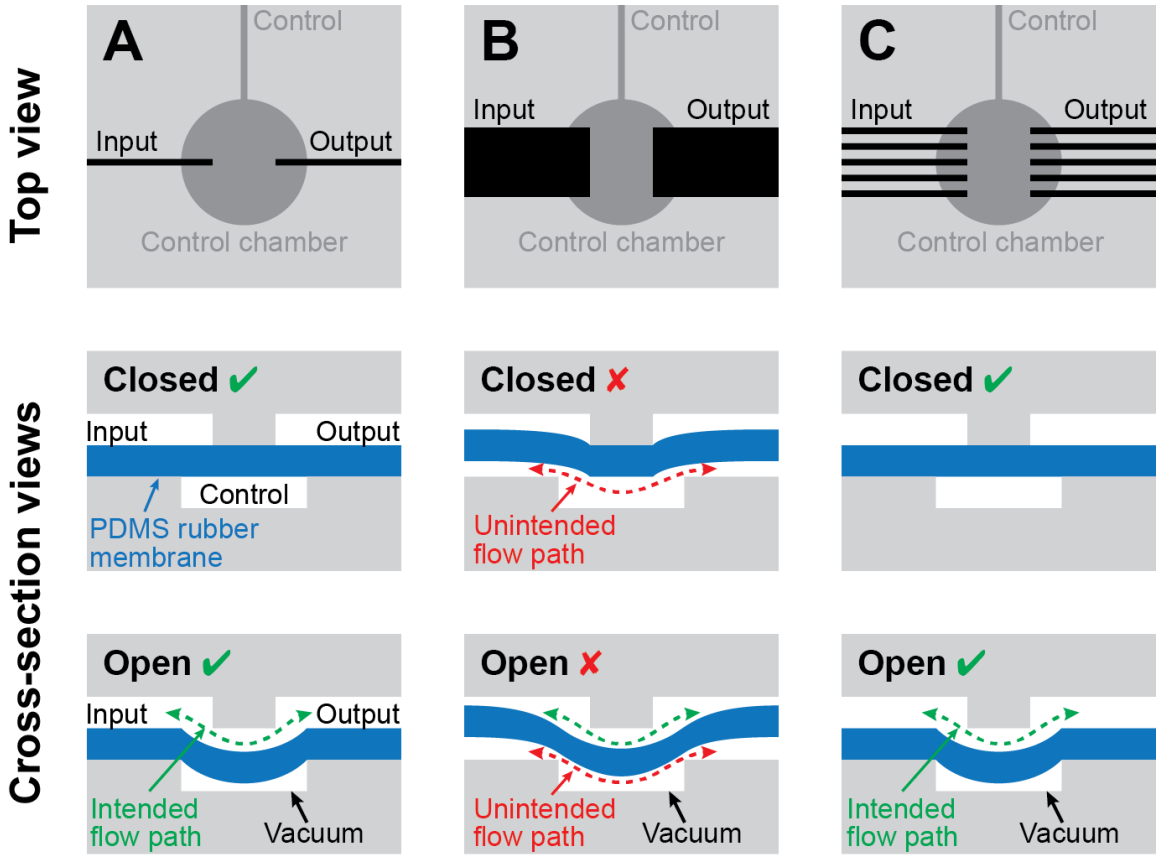

Figure 2.2: Adapting normally-closed monolithic membrane valves for soft robotic applications. **(A)** In conventional microfluidic monolithic membrane valves [25], fluid flow between the Input channel and the Output channel is blocked by a flexible silicone membrane (blue). When a vacuum is applied to the Control channel, the silicone membrane is pulled into the Control chamber, opening a path for flow between the Input and Output channels (green dashed arrow). Though adequate for microfluidic applications, the rate of air flow through conventional monolithic membrane valves is too small for controlling larger and faster-moving soft robots. **(B)** Increasing the widths of the Input and Output channels significantly increases the rate of flow through the valve, but this change brings an undesirable side effect: the silicone membrane can stretch into the Input and Output channels as if they were Control chambers, creating unintentional flow paths (red dashed arrows). **(C)** Using multiple narrow-width Input and Output channels in parallel eliminates the risk of unintentional flow paths while still maintaining a high flow rate suitable for use controlling soft robotic actuators.

a history of use in complex pneumatic logical circuits for controlling microfluidic devices. For example, the authors and other researchers have demonstrated microfluidic pneumatic versions of Boolean logic gates [34], binary memory [24], mathematical calculators [34], clocks [19], programmable biochemical processors [33, 36, 35, 39, 46], and even simple computers (finite state machines [3]) [59, 18].

However, in their conventional form, these pneumatic logic circuits would only be capable of controlling small and slow-moving robots. This is because the small microfluidic-scale channels in these circuits can only accommodate small volumes of air at low flow rates. So while conventional normally-closed valve-based pneumatic logic circuits have been proposed for controlling soft robots [50], experimental demonstrations of these control systems are limited.

To enable monolithic membrane valve-based logic circuits to control larger and faster-moving robots, we first tried increasing the size of the channels in these circuits. As the cross-sectional area of a channel gets larger, its capacity for flow increases dramatically. However, when we tested pneumatic logic circuits with larger channels, we found a fundamental problem with these designs: as shown in Fig 2.2B, when the widths of the Input and Output channels are increased, the silicone rubber membrane can be pulled or pushed into the Input or Output channels, creating an unintentional new path for air flow on the other side of the membrane (red dashed arrows in Fig 2.2B). In this manner, the Input and Output channels unintentionally behave like Control chambers, and the pneumatic logic circuit no longer functions as intended.

We then developed a modified pneumatic logic circuit design that handles larger

air flow rates without compromising the circuit's functionality. We accomplished this by using multiple channels in parallel everywhere that high flow is needed. When multiple parallel channels are used as the Input and Output channels in a valve (Fig 2.2C), the resulting "high-flow" monolithic membrane valve can carry several times more flowing air when open and still function correctly in pneumatic logic circuits. Consequently, high-flow monolithic membrane valves can control larger and faster-moving soft robots than their traditional microfluidic counterparts.

2.3.2 "Truth table" for high-flow monolithic membrane valves

Next, we needed to define the rules for how high-flow monolithic membrane valves behave in pneumatic logic circuits. While monolithic membrane valves can be thought of as analogous to transistors in electronic circuits, the physics of flowing air in pneumatic logic is fundamentally different from the physics of flowing electrons in electronic logic. In particular, pneumatic logic circuits built from normally-closed valves are capable of storing or trapping pressure differentials inside the circuit; this is fundamentally different from electronic circuits which usually require an additional component (like a capacitor or a floating-gate transistor) to store a charge (the electronic analog of a pressure).

A monolithic membrane valve has three connections (Input, Control, and Output, as shown in Fig 2.2), each of which can receive either vacuum (abbreviated V) or atmospheric pressure (abbreviated A). This results in eight possible states for the valve (abbreviated in the order *Input Control Output*): AAA, AAV, AVA, AVV, VAA, VAV, VVA, and VVV. We describe the state of the valve during each of these eight states in the

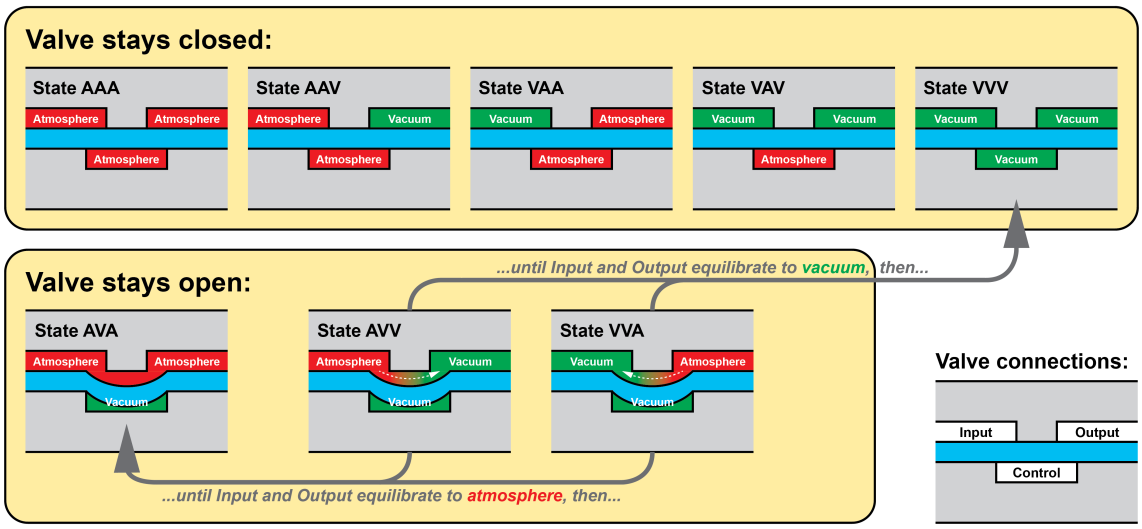

Figure 2.3: A "truth table" showing cross-sections of a high-flow monolithic membrane valve during each of the eight possible combinations of atmospheric pressure (A) and vacuum (V) applied to the valve's three connections (Input, Control, and Output). When atmospheric pressure is applied to all three connections on a valve, the valve has equal pressure applied on both sides of the flexible membrane, so the membrane remains in its resting (closed) state (AAA). If the Control connection on a valve receives atmospheric pressure, then the valve will remain closed, regardless of whether the Input and Output connections receive vacuum or pressure (AAV, VAA, and VAV). When vacuum is applied to all three connections on a valve, the valve has the same pressure on both sides of the flexible membrane, so the membrane remains at rest and the valve stays closed (VVV). If vacuum is applied to the Control connection while the Input and Output connections are at atmospheric pressure, then the pressure differential across the flexible membrane causes the membrane to stretch into the Control chamber and opens the valve (AVA). When vacuum is applied to the Control connection and the Output connection while the Input connection is at atmospheric pressure (AVV), the valve opens and air flows from the Input to the Output. Likewise, when vacuum is applied to the Control and Input connections while the Output is at atmospheric pressure (VVA), the valve opens and air flows from the Output to the Input. In both cases, the valve remains open and air continues to flow as long as there is a pressure difference between the Input and Output connections. *However, if the pressures at the Input and Output connections equalize (both become vacuum, or both become atmospheric pressure), the valve will automatically transition to a new state.* If both the Input and the Output reach atmospheric pressure, then the valve will transition to state AVA and remain open. But if both the Input and the Output reach vacuum, then the valve will transition to state VVV and automatically close. This automatic transition from state AVV or VVA to state VVV is particularly useful because it can "trap" a vacuum in a region of a pneumatic logic circuit; the vacuum remains trapped until it is vented by opening a path to atmospheric pressure using state AVV or VVA again. This serves as a one-bit nonvolatile pneumatic "memory": a section of channel represents 1 (TRUE) if it contains a trapped vacuum and 0 (FALSE) if it contains atmospheric-pressure air.

"truth table" shown in Fig 2.3.

2.3.3 Design of a pneumatic eight-bit random-access memory

As a proof-of-concept for using high-flow monolithic membrane valves to control soft robots, we designed, fabricated, and tested a pneumatic eight-bit random-access memory (RAM). This pneumatic logic circuit can set and "remember" the air pressure level at eight outputs in the circuit. By connecting soft robotic actuators to these outputs, we can use the pneumatic RAM to control eight independent actuators.

The equivalent functional logic diagram for our pneumatic eight-bit RAM is shown in Fig 2.4. The pneumatic RAM is controlled by four computer-controlled solenoid valves, shown in orange in Fig 2.4. Three of these solenoid valves provide the values of three Address bits, which are used to select which Memory bit in the pneumatic RAM to set, and the fourth solenoid valve provides the value of the Data bit which is stored in the selected Memory bit. This is the only off-chip control hardware required to operate the pneumatic RAM; all of the remaining logic operations in Fig 2.4 (blue box) are performed by the pneumatic RAM chip. The pneumatic RAM chip's logical operations include a pneumatic demultiplexer that connects the value of the Data bit to the Memory bit selected by the Address bits, and eight D-type flip-flops which maintain the value of each Memory bit between setting events. Finally, each of the eight Memory bits can be connected to a soft robotic actuator, which is operated by the pressure or vacuum stored in the Memory bit. In this manner, our pneumatic RAM uses four computer-controlled solenoid valves to control eight independent soft robotic actuators. In general, this approach can control 2^{n-1}

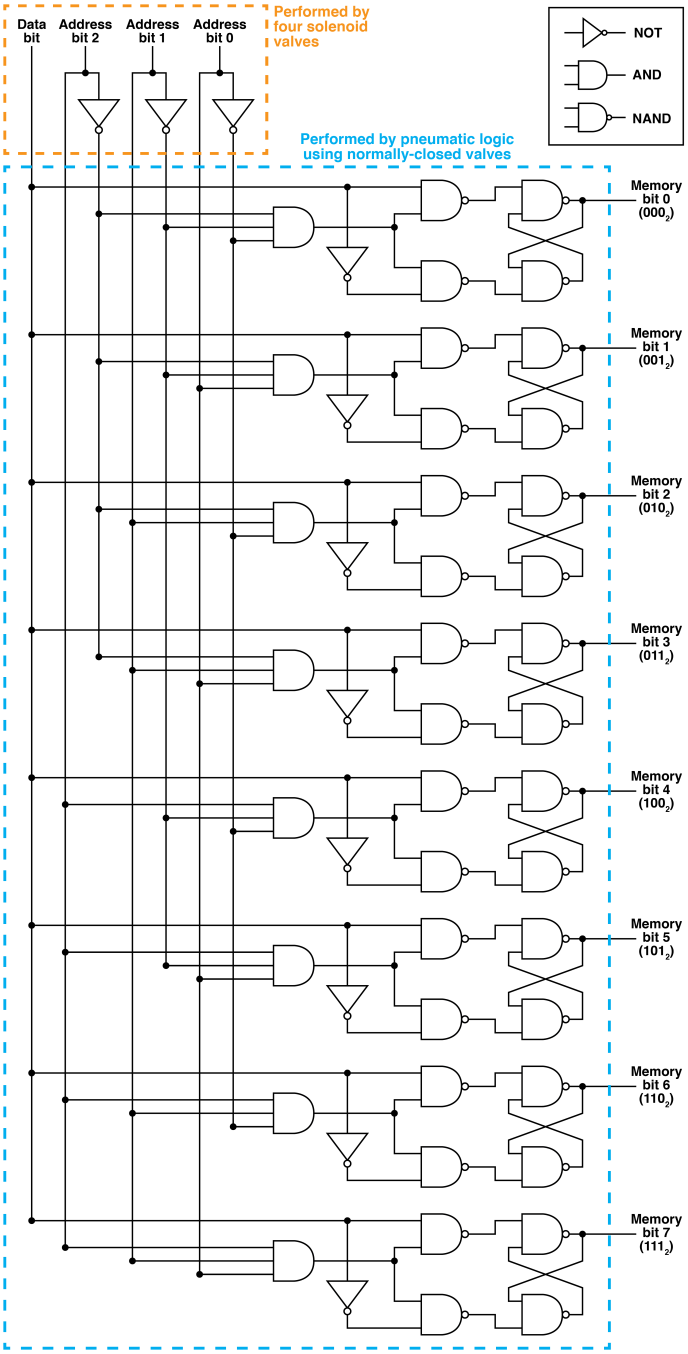

Figure 2.4: Logic diagram for our pneumatic eight-bit random-access memory. Four solenoid valves (orange region in the diagram) provide three pneumatic Address bits (for selecting which Memory bit to set) and one Data bit (for setting the value of the selected Memory bit); the rest of the logic diagram (blue region) is performed by a pneumatic logic circuit. Eight outputs (labeled Memory bits 0 through 7) provide access to the eight stored pressure levels in the pneumatic RAM and are connected to eight fingers in two soft robotic hands.

independent actuators using n computer-controlled solenoid valves.

The physical layout of the pneumatic eight-bit RAM chip is shown in Fig 2.5 (design file available as online *Supplementary Information*). The pneumatic RAM chip contains 14 high-flow monolithic membrane valves arranged in a pneumatic logic circuit that is functionally equivalent to the blue region in the logic diagram in Fig 2.4. The fact that only 14 monolithic membrane valves are needed to perform the functions of a logic diagram containing 48 logic gates is remarkable; it is a direct consequence of the normally closed nature of these valves, which enables a *single valve* to trap and "remember" a pneumatic signal and therefore function as a five-gate D-type flip-flop.

2.3.4 Operation of the pneumatic eight-bit RAM

In Fig 2.6 and the following paragraphs, we step through the process of using the pneumatic RAM chip to contract and extend several soft robotic finger. Additionally, a table showing the state of each valve at each step and a video recording of a pneumatic RAM chip during operation are available as online *Supplementary Information*.

The pneumatic RAM uses atmospheric-pressure air to represent a "0" or FALSE value, and vacuum to represent a "1" or TRUE value. The soft robotic fingers are extended when connected to atmospheric pressure and contracted when connected to vacuum. Initially, all of the channels inside the pneumatic RAM are at atmospheric pressure. This means that the eight stored pneumatic memory bits default to atmospheric pressure (all 0 or FALSE), and the pneumatic RAM currently stores the value 00000000_2 (the subscript "2" indicates that this number is in base-2 or binary). The eight soft robotic fingers connected

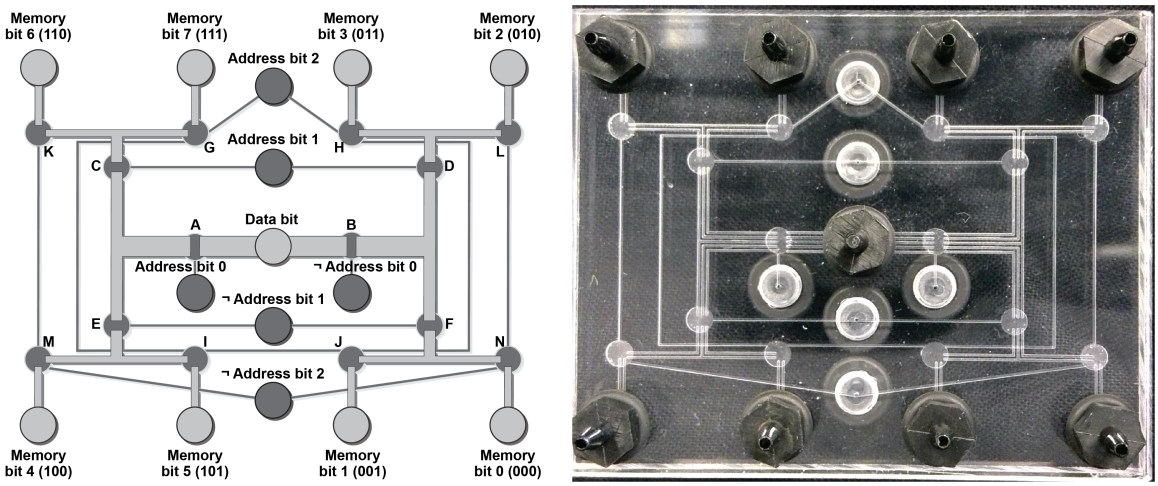

Figure 2.5: Design and photograph of a pneumatic eight-bit RAM for controlling soft robotic fingers. The 6.35 cm × 5.08 cm chip includes eight ports for connecting fingers to the Memory bits, a single Data bit connection that sets the state (contracted or extended) of each finger, three Address bit connections (and their negations, marked with "¬") which are used to select which Memory bit (and therefore which finger) to set, and 14 pneumatic valves (labeled A through N) that execute the logical functions shown in the blue box in Fig 2.4.

to these bits are also under atmospheric pressure and are therefore initially all extended.

Step 1: Contract Finger 2

In Step 1 of Fig 2.6 we use the pneumatic eight-bit RAM to contract Finger 2. This finger is connected to Memory bit 2, which has the address 010_2; this means that to select Finger 2, we need to set Address bit 0 to 0 (atmosphere), Address bit 1 to 1 (vacuum), and Address bit 2 to 0 (atmosphere). The negated Address inputs (marked with a "¬" or NOT sign) automatically receive the opposites of these values, so ¬ Address bit 0 is set to 1 (vacuum), ¬ Address bit 1 is set to 0 (atmosphere), and ¬ Address bit 2 is set to 1 (vacuum). All of these Address bit values are supplied to the pneumatic RAM using three solenoid valves. Once the Address bits have been set to address Finger 2, vacuum is applied

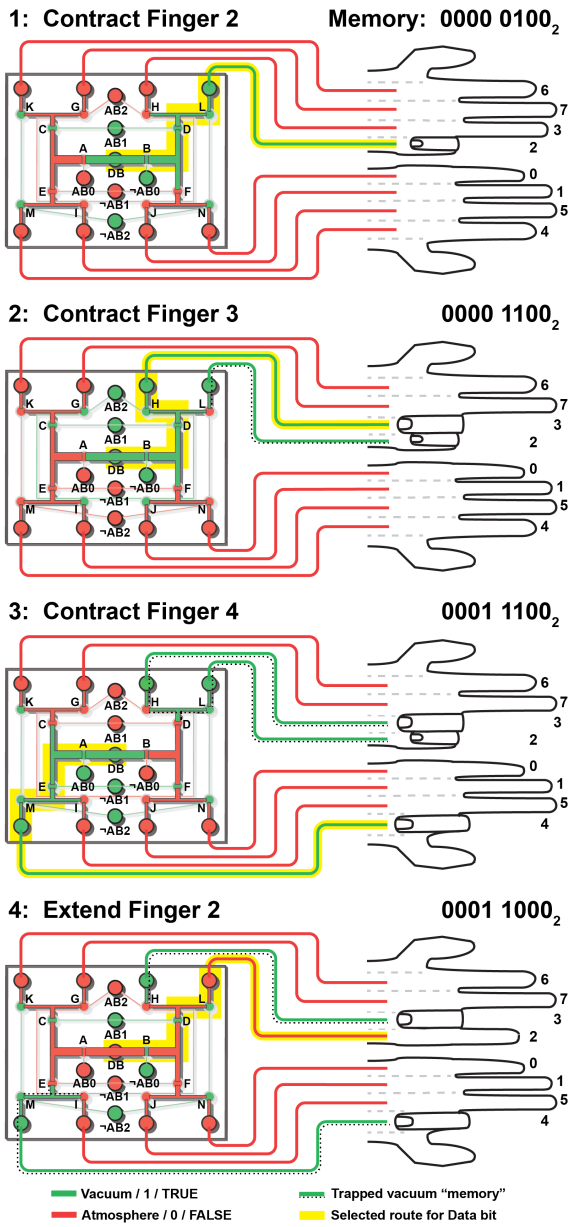

Figure 2.6: Pressure inside each channel (either green for vacuum, or red for atmospheric pressure) while using the pneumatic eight-bit RAM to contract and extend several soft robotic fingers. In each step, the vacuum or pressure applied to the Address bits (AB0, AB1, AB2, and their negations ¬AB0, ¬AB1, ¬AB2) determine the route (highlighted in yellow) followed by the Data bit (DB) signal to the selected finger. Channels containing trapped vacuum "memories" are marked using a dotted line, and the binary representation of the contents of the pneumatic RAM is shown. A detailed explanation of each step is provided in the text, and a table showing the state of each valve (A through N) during each step and a video recording of the chip during operation are available as online *Supplementary Information*.

to the single Data connection in the center of the pneumatic RAM chip. This vacuum follows a path determined by the valves opened by the Address inputs, first through Valve B (which is held open by the vacuum applied to ¬ Address bit 0 according to State *VVA* in Fig 2.3), then through Valve D (which is held open by the vacuum applied to Address bit 1; State *VVA*), then finally through Valve L (which is held open by the vacuum applied to ¬ Address bit 2; State *VVA*). The Data vacuum then reaches Finger 2. To illustrate the logic levels inside the device, in Fig 2.6 (and in subsequent figures) we colored channels and tubes containing atmospheric pressure air (FALSE or 0) in red and vacuum (TRUE or 1) in green. The route followed by the vacuum from the Data bit to Memory bit 2 and Finger 2 is highlighted yellow in Fig 2.6 Step 1.

Finger 2 was initially filled with air at atmospheric pressure. When the pneumatic RAM routes vacuum from the Data bit to Memory Bit 2 using the process described above, this vacuum pulls air out of the finger and lowers the air pressure inside the finger; this causes the finger to contract as desired. As the air pressure inside the finger drops, it ultimately reaches a vacuum level equal to the vacuum used to control the pneumatic RAM. At this point, Valve L is no longer in State *VVA*; rather, it now is in State *VVV*. As explained in Fig 2.3, in State *VVV* a valve no longer has a pressure differential holding it open, so the valve closes. This transition of Valve L from open (*VVA*) to closed (*VVV*) happens automatically after the air pressure inside the finger drops to the vacuum level, and it also happens in Valves D and B, which also close automatically. In this manner, the vacuum causing Finger 2 to contract is now trapped inside the finger, separated from the rest of the pneumatic RAM chip by valves that closed automatically when the finger was done

contracting. At this point, the pneumatic RAM chip now stores the value 0000 0100_2.

Close inspection of Fig 2.6 reveals that in addition to the valves (B, D, and L) that are opened in Step 1 to create a path for vacuum to flow to Finger 2, four other valves also open (Valves C, K, M, and N; due to State AVA). However, seven other valves remain closed (Valves A, F, and H due to State VAA or AAV; and valves E, G, I, and J due to State AAA). These seven closed valves block the flow of air through the four open valves, so the only path for vacuum to flow through the chip is the intended path (through valves B, D, and L, to Finger 2). (As an aside, we acknowledge that our terminology of a "flowing vacuum" is nonstandard, but we nevertheless find it helpful to sometimes describe a pneumatic signal as a "vacuum flowing from A to B" rather than "air flowing from B to A.")

Step 2: Contract Finger 3

Next, in Step 2 of Fig 2.6 we wish to contract Finger 3 while keeping Finger 2 contracted (in other words, we need to transition the value stored by the pneumatic RAM from 0000 0100_2 to 0000 1100_2). To accomplish this, we apply the address of Memory bit 3 (011_2) to the Address bits: Address bit 0 = 1 (vacuum), Address bit 1 = 1 (vacuum), and Address bit 2 = 0 (atmosphere), along with their opposites to the negated Address bits: ¬ Address bit 0 = 0 (atmosphere), ¬ Address bit 1 = 0 (atmosphere), and ¬ Address bit 2 = 1 (vacuum). This opens a path (colored yellow in Fig 2.6 Step 2) for vacuum to flow from the Data connection, through Valves B, D, and H (all held open in state VVA), to Finger 3 which then contracts. As in Fig 2 above, when the contents of Finger 3 reach vacuum, Valves B, D, and H all close due to the automatic transition from State VVA to

State *VVV*.

What keeps the original Finger 2 contracted (under vacuum) while the pneumatic RAM chip is setting the new Finger 3? As shown in Step 2, Valve L (which leads to Finger 2) is in State *VAV*, with vacuum from the pneumatic RAM chip's Data bit applied to the valve's Input, atmospheric pressure applied to the valve's Control, and the vacuum trapped inside the finger present at the valve's Output. State *VAV* dictates that this valve will remain closed, so the vacuum remains trapped inside Finger 2 and the channels and tubing leading up to it (this trapped vacuum "memory" is marked with a dotted line in Fig 2.6 Step 2). This trapped vacuum keeps Finger 2 contracted, even while the pneumatic RAM is contracting Finger 3. At the conclusion of this step, the the pneumatic RAM now stores the value $0000\ 1100_2$.

Step 3: Contract Finger 4

Next, in Step 3 of Fig 2.6, we wish to contract Finger 4 while keeping Fingers 2 and 3 contracted. This corresponds to transitioning the value stored in the pneumatic RAM from $0000\ 1100_2$ to $0001\ 1100_2$. As before, we apply the address of Memory bit 4 (100_2) to the pneumatic RAM's Address bits and the negated Address bits. This opens a path (colored yellow) for vacuum to flow from the Input connection, through valves A, E, and M (held open in state *VVA*), to Finger 4 which then contracts. Again as before, Valves A, E, and M then automatically close (transitioning from *VVA* to *VVV*) when the contents of Finger 4 reach vacuum.

At this point, Fingers 2 and 3 both need to remain contracted while the pneumatic RAM chip is contracting Finger 4. Inspection of Step 3 in Fig 2.6 shows how this is possible.

Valve L, which was previously closed and used to trap the vacuum inside Finger 2, is actually *open* in Step 3 because it shares a Control line with Valve M (which had to be opened to send vacuum to Finger 4). However, Finger 2 remains contracted because the valves upstream of Valve L in the pneumatic RAM, Valves H and D, are both closed. Thus, while a small amount air from the channels between Valves L, H, and D does flow toward the vacuum trapped in Finger 2 in Step 3, the volume of this air is negligible compared to the volume of vacuum trapped inside Finger 2, so the vacuum inside Finger 2 remains virtually unchanged and the finger remains contracted. Similarly, the vacuum inside Finger 3 remains trapped by Valve H being closed in Step 2. At the end of this step, the pneumatic RAM's contents are $0001\ 1100_2$.

Step 4: Re-extend Finger 2

Finally, in Step 4 of Fig 2.6, we wish to re-extend Finger 2 while keeping Fingers 3 and 4 contracted (that is, transition the pneumatic RAM's contents from $0001\ 1100_2$ to $0001\ 1000_2$). To extend Finger 2, the pneumatic RAM chip needs to route atmospheric pressure to the finger; the resulting air flow into the finger destroys the trapped vacuum and resets the finger to its resting (extended) state. To accomplish this, we again apply the address of Memory bit 2 to the pneumatic RAM's Address bits and negated Address bits. This opens a path (colored yellow) for atmospheric pressure to flow from the Data bit connection, through valves B, D, and L (now held open in State *AVV*), to Finger 2, thereby extending the finger.

Once more, inspection of Step 4 reveals why Fingers 3 and 4 remain contracted while the pneumatic RAM extends Finger 2. Finger 3's trapped vacuum is sealed by Valve

H, which is closed in Step 4. Finger 4's trapped vacuum is no longer sealed by Valve M, which was opened in Step 4 as a consequence of opening Valve L which shares a Control line with Valve H; however, the neighboring Valves I and E *are* closed in Step 4, and those valves seal the trapped vacuum inside Finger 3 during Step 4. The pneumatic RAM now stores the value 0001 1000_2.

In this manner, the pneumatic RAM can set soft robotic fingers to any desired combination of contracted or extended states, and the fingers will "remember" their state until they are set to a different state.

2.3.5 Testing the pneumatic RAM

After fabricating our pneumatic RAM, we made eight 3D-printed soft robotic fingers (details in *Materials and Methods* below) to use in testing the pneumatic RAM. These flexible elastomer fingers are normally extended when their hollow interiors are under atmospheric pressure. When a vacuum is applied to a finger, the finger contracts and curls into a C-shape. Restoring atmospheric pressure to the finger causes it to extend again. We used these fingers in a series of tests to confirm that the pneumatic RAM operates as intended when controlling a soft robot.

In the first phase of this testing, we used the pneumatic RAM to contract one finger at a time while holding the other fingers extended. Frames from a video recording of the experiment are shown in Fig 2.7. As the pneumatic RAM steps through all eight possible values for the three Address bits (from 000_2 for Memory bit 0, to 111_2 for Memory bit 7), the finger connected to each Memory bit contracts, thereby confirming that the Data

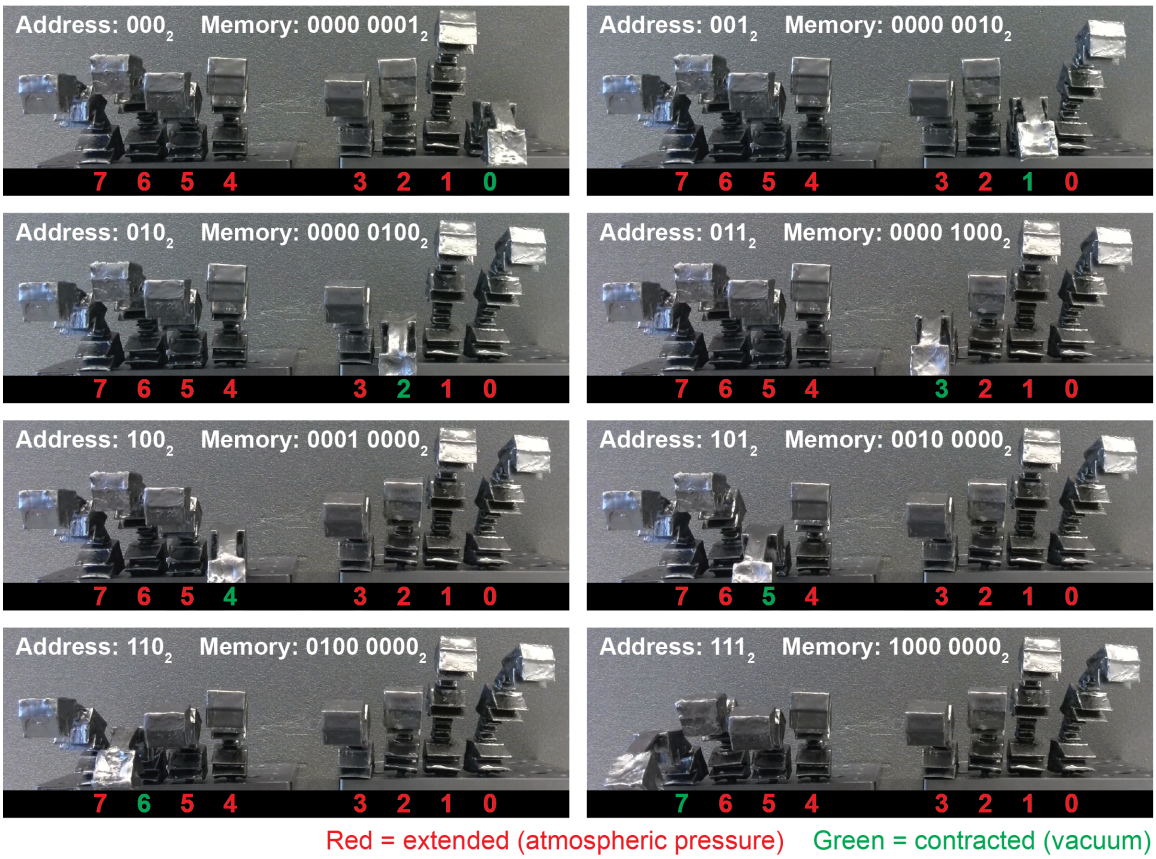

Figure 2.7: Video frames from using the pneumatic eight-bit RAM to contract eight soft robotic fingers one-at-a-time. The pneumatic RAM cycles through all eight addresses from 000_2 to 111_2 to set the pneumatic RAM's memory to the values shown (from 0000 0001_2 to contract just Finger 0, to 1000 0000_2 to contract just Finger 7).

bit vacuum was successfully routed to each finger in turn.

After demonstrating that the pneumatic RAM chip can control each finger independently, we then tested whether the pneumatic RAM can set and maintain all eight fingers in any desired pattern of contracted and extended fingers. This is a considerably more difficult task for the pneumatic RAM chip: there are 2^8 or 256 different patterns of contracted or extended fingers, ranging from all eight extended (0000 0000_2) to all eight contracted (1111 1111_2) and every combination in between, and each finger must "remember"

its state (contracted or extended) while other fingers are being set.

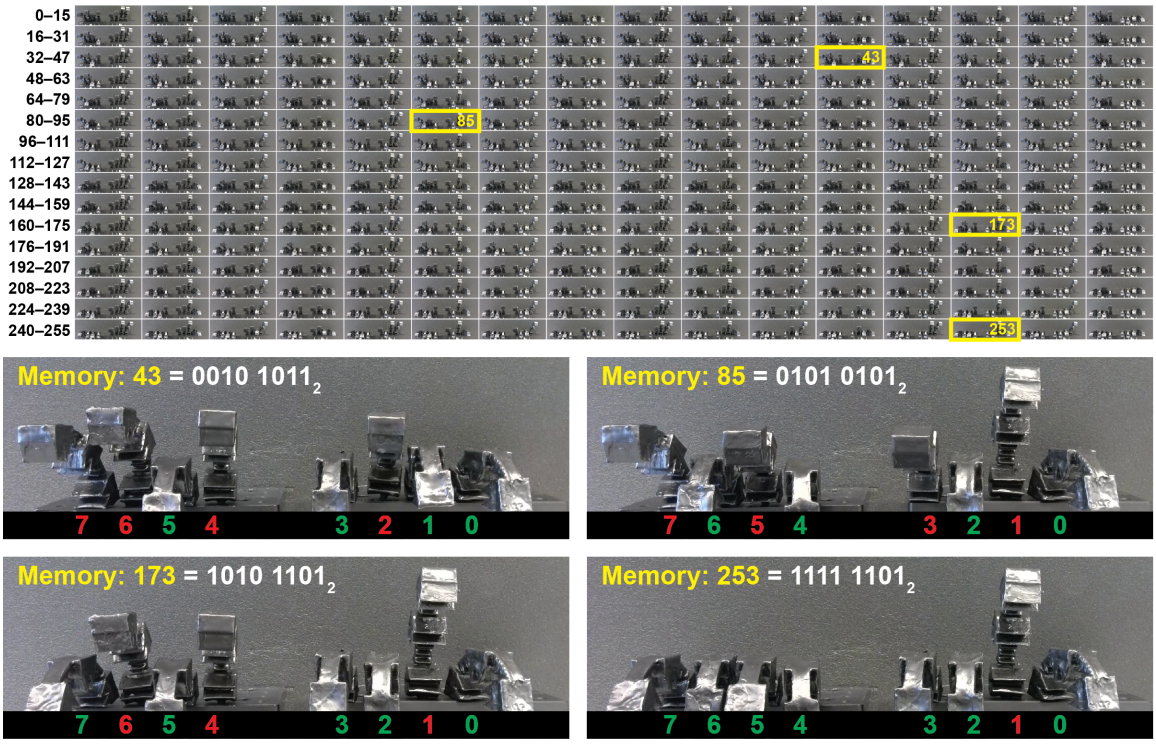

Figure 2.8: Video frames from using the pneumatic RAM to set eight soft robotic fingers to all 256 different possible patterns, ranging from all extended (corresponding to a value of 0000 0000_2 or 0 stored in the pneumatic RAM) to all contracted (a value of 1111 1111_2 or 255 stored in the pneumatic RAM), with closeups of the fingers while the pneumatic RAM is storing the values 43, 85, 173, and 253. Closeups of all 256 frames are available as *Supplementary Information*.

Fig 2.8 shows frames from a video recording of the pneumatic RAM setting all eight fingers to all 256 possible patterns. The video starts with all eight fingers extended, corresponding to a value of 0000 0000_2 (or 0 in decimal) stored in the pneumatic RAM. Next, the pneumatic RAM uses the address of Memory bit 0 (000_2) to route a vacuum that contracts Finger 0. This transitions the value stored by the pneumatic RAM to 0000 0001_2 (or 1 in decimal). In the next step, the pneumatic RAM uses the address of Memory bit 1 (001_2) to route a vacuum that contracts Finger 1, then uses the address of Memory

bit 0 again (000_2) to contract Finger 0; this transitions the pneumatic RAM contents to 0000 0010_2 (or 2 in decimal). This process is continued in a binary counting pattern—0000 0011_2 (or 3 in decimal), 0000 0100_2 (or 4 in decimal), 0000 0101_2 (or 5 in decimal), and so on—through all 256 possible states of the pneumatic RAM contents, all the way to 1111 1111_2 (255 in decimal). Photographs of the fingers in all 256 different memory states are shown in Fig 2.8, along with closeups corresponding to the pneumatic RAM storing the values 43 (0010 1001_2), 85 (0101 0101_2), 173 (1010 1101_2), and 253 (1111 1101_2). In addition, closeups of the fingers during all 256 different states of the memory contents are available as *Supplementary Information*. No errors were observed during the experiment. This confirms that the pneumatic RAM chip can remember 256 different values and use these values to set and maintain eight soft actuator fingers according to any of 256 different actuation patterns.

2.3.6 Characterizing pneumatic "memory"

As noted above, the pneumatic RAM can control 2^{n-1} independent soft robotic actuators using n computer-controlled solenoid valves. This exponential relationship means that an extremely large number of actuators can be controlled by a modest amount of control hardware. But since the pneumatic RAM can only update the state of one actuator at a time, as the number of actuators grows, the amount of time between updates for a given actuator also increases. Limited update frequency is not an issue for an actuator being held at atmospheric pressure (FALSE or 0) because atmospheric pressure is the normal at-rest state in our pneumatic RAM chip; it can maintain actuators at atmospheric pressure indefinitely. However, update frequency *is* important for an actuator being held under

vacuum (TRUE or 1) because small unintended air flows (due to *e.g.* small volumes of atmospheric-pressure air routed to the actuator during routine pneumatic RAM operation as described above) could deplete the trapped vacuum over time. Consequently, the trapped vacuum that "remembers" the state of an actuator needs to last for as long as possible, at least long enough to maintain the state of an actuator until the pneumatic RAM refreshes the vacuum during its next cycle of setting the actuators.

To determine how long a latched vacuum can hold a soft robotic finger in the contracted state, we first developed a method for measuring the amount of finger contraction. Our 3D-printed soft robotic fingers normally have a slight curvature when extended; this curvature gives rise to the angle θ_0 shown in Fig 2.9A. When a vacuum is applied to the finger, the finger contracts, increasing its curvature by an additional angle $\Delta\theta$ as shown in the inset images in Fig 2.9B. To measure and monitor the degree of contraction over time in a finger containing a latched vacuum, we wrote a MATLAB script (available as online *Supplementary Information*) that extracts θ_0 and $\Delta\theta$ from a video recording of the side view of the finger.

We then wrote a program for the pneumatic RAM that subjects a finger to a "stress test" intended to determine how long one finger can remain contracted by a trapped vacuum from a Memory bit while the pneumatic RAM is actively controlling other fingers. This program initially routes a vacuum to a finger for 60 seconds, during which time the finger contracts and the vacuum is sealed inside the finger when the pneumatic RAM chip's valves automatically close, as described above. Applying vacuum for an excessively-long 60 seconds ensures that the finger is fully contracted. Then, the program uses the pneumatic

RAM to continuously contract and extend the other three fingers connected to the IC. This is a "stress test" because, while the pneumatic RAM is applying vacuum and atmospheric pressure to the other fingers, small amounts of atmospheric-pressure air from inside the pneumatic RAM chip's channels are leaked to the trapped vacuum in the first finger (this is a normal side-effect of pneumatic RAM operation, as explained in the description of Fig 2.6 above). If enough atmospheric-pressure air enters the first finger, it could degrade or even eliminate the trapped vacuum in the Memory bit that is holding the finger contracted, effectively causing the pneumatic RAM to "forget."

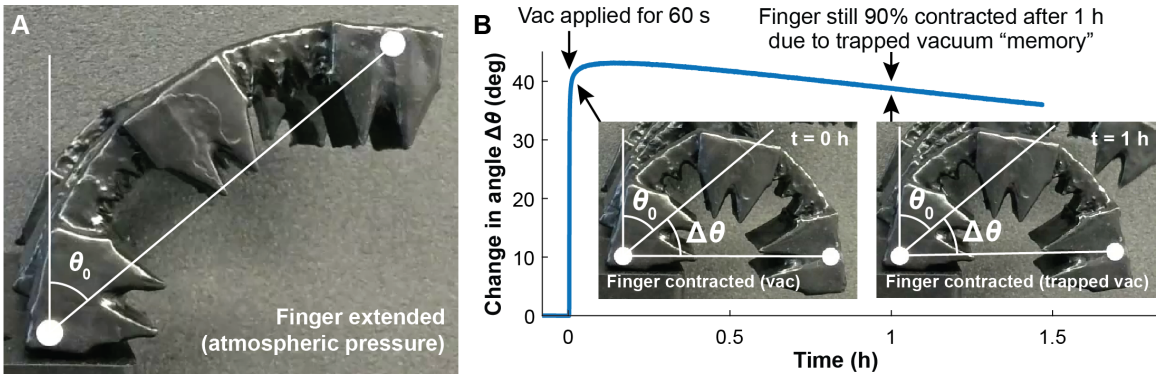

Figure 2.9: Measuring the amount of time that the pneumatic RAM can "remember" the value of a Memory bit (and how long a soft robotic finger connected to the Memory bit can maintain its actuation state). **(A)** At atmospheric pressure, the finger has a natural curve with an angle θ_0. **(B)** At time = 0, the pneumatic RAM applies vacuum from the Data input to the Memory bit and the finger for an excessively-long 60 s, which causes the finger to contract an additional $\Delta\theta$ degrees. After 60 s the pneumatic RAM disconnects the Data vacuum from the Memory bit, but the finger remains contracted due to the vacuum trapped in the Memory bit and inside the finger. For the next hour, the pneumatic RAM cycled through three other fingers, contracting and extending one every ten seconds. Even though the pneumatic RAM chip's operation introduces small amounts of atmospheric-pressure air to the first finger, the finger nonetheless remained 90% contracted 1 h after the applied vacuum was removed. Therefore, a trapped vacuum at a pneumatic RAM Memory bit can hold a finger contracted for at least an hour before needing to be "refreshed" by the pneumatic RAM. A closeup view of the first few seconds of this plot is available as online *Supplementary Information.*

A typical result obtained from subjecting a soft robotic finger to this pneumatic memory "stress test" is shown in Fig 2.9B. The slow decrease in the measured contraction angle $\Delta\theta$ confirms that some atmospheric-pressure air is entering the trapped vacuum inside the finger while the pneumatic RAM chip is operating other fingers. However, the finger is still 90% contracted one hour after the pneumatic RAM routed the original vacuum to the finger. This was the median result of the three soft robotic fingers we used in this test; in the other experiments (available in online *Supplementary Information*) the pneumatic RAM kept the fingers at least 90% contracted for 22 minutes and 1.2 hours. These results show that the Memory bits on the pneumatic RAM can maintain the state of a soft robotic finger for extended periods of time before needing to be "refreshed."

A closeup of the first few seconds of the plot in Fig 2.9B (available as online *Supplementary Information*) shows that the finger is 70% contracted after just 4 seconds, 80% in 7 seconds, and 90% contracted in 18 seconds. For many applications, 70% contraction of an actuator may be adequate, so each actuator could be set in 4 seconds and remember its state for at least an hour. In this manner, a pneumatic RAM could control 900 independent actuators, updating the state of each actuator every hour. To control this massive number of actuators, the pneumatic RAM would require only $\log_2 900 + 1 = 11$ solenoid valves (ten to set the ten Address bits and their negations, and one to set the Data bit); this represents a *99% reduction* in the amount of control hardware that would normally be required to operate 900 independent soft robotic actuators. For applications that require more frequent updates of actuator state, one can halve the number of independent actuators and double the update frequency. So, a pneumatic RAM could set 450 actuators to

any desired actuation pattern every 30 minutes, or set eight actuators to any pattern every minute, and so on.

Note that the projections above assume that the pneumatic RAM sets every actuator during each operation cycle. This is not strictly required, and for many applications it would be advantageous to set some actuators more frequently than others. In this manner, a pneumatic RAM could support both rapidly operating actuators (updated every few seconds) and less-frequently operated actuators (updated only when needed) on the same robot. Additionally, the subset of actuators that require the most rapid operation can always be connected directly to solenoid valves as is typically done, with the remaining actuators controlled via a pneumatic RAM.

2.3.7 Pneumatic digital-to-analog conversion

The analysis above treats our soft robotic fingers as binary (either contracted or extended). However, it is noteworthy that the pneumatic RAM can also maintain these fingers at any intermediate position. This can be accomplished by using the pneumatic RAM to set the pressures at the Memory bits to intermediate values between full atmospheric pressure (corresponding to full extension) and full vacuum (corresponding to full contraction). Specifically, by changing the amount of time that the pneumatic RAM applies full vacuum or full atmospheric pressure from the Data input to a Memory bit, the pneumatic RAM can set an arbitrary pressure in Memory bit and therefore set the connected actuator to any desired angle. For example, having the pneumatic RAM deliver vacuum to a Memory bit for about 2.5 s duration sets the associated finger about halfway between the contracted and extended states (see *Supplementary Information*). In this man-

ner, the pneumatic RAM can function like a multi-channel digital-to-analog converter and data buffer [43], setting multiple soft robotic actuators to any desired position (analog) using different-duration pulses of a constant vacuum or atmospheric pressure (digital), and maintaining the actuators in those positions for an extended period of time. The accuracy and precision of this pneumatic digital-to-analog conversion would depend upon the actuators having predictable and consistent behavior, so this method may not be suitable for precision control, but it nonetheless demonstrates that the pneumatic RAM is not strictly limited to just binary (either contracted or extended) control of actuators.

2.3.8 Simultaneous actuation of multiple fingers

In the tests described above, our pneumatic RAM was limited to changing the state of one soft robotic finger at a time. Since the fingers can "remember" their state thanks to trapped pressures in the pneumatic RAM's Memory bits, this one-at-a-time operation does not limit the number of finger actuation patterns that are possible when multiple fingers are controlled by the same pneumatic RAM. However, there are many situations where it would be desirable to change two or more fingers *simultaneously*. For example, a biomimetic soft robotic hand might contract all five of its fingers simultaneously to grasp an object, then extend all of its fingers simultaneously to release it. If two or more actuators are *always* to be operated in lock-step, then they could be connected to the same Memory bit on the pneumatic RAM; this would guarantee true simultaneous actuation at the expense of individual control. However, for applications that require both individual *and* simultaneous control of actuators, a different solution is needed.

To solve this problem, we again turned to techniques originally developed for

electronic circuits. In this case, the principle of *time-division multiplexing* (TDM) allows electronic circuits to send multiple messages simultaneously through a single communication channel [43]. In TDM, a switch continuously and rapidly alternates between the original signals, sending only part of each message at a time. When these parts are reconstructed on the other end of a communication channel, the full messages are recreated.

Inspired by electronic TDM, we hypothesized that by operating the pneumatic RAM at high speed, we could use it to send brief pulses of atmospheric pressure or vacuum to each Memory bit and each connected finger. The duration of each pulse would be short, too short for a single pulse to fully contract or extend a finger. However, as multiple pulses are delivered to each finger, their effects would accumulate and ultimately contract or extend the fingers. By cycling through the fingers rapidly in a manner akin to electronic TDM, the pneumatic RAM could contract or extend several fingers essentially simultaneously.

To test this idea, we needed a task for a soft robot that requires multiple fingers to be actuated simultaneously. We decided that playing a piano keyboard was a suitably complex task because piano playing requires pressing not only individual keys but also multiple keys at once to play chords. The sound produced when a key is pressed indicates that the attempt to press the key has been successful.

As shown in Fig 2.10, we mounted a set of four soft actuator fingers in front of an electric piano keyboard and wrote a program that makes the pneumatic RAM constantly deliver brief pulses of vacuum from the Data input to Memory bits 1, 2, and 4, which in turn were connected to three of the four fingers (the fourth finger received no vacuum pulses). The amount of time the pneumatic RAM connected each Memory bit to vacuum was varied

from 150 ms down to only 40 ms. With three fingers updated in each cycle of the pneumatic RAM, these times correspond to cycle times ranging from 450 ms to 120 ms, meaning that each finger receives from 2.2 to 8.3 vacuum pulses per second.

Fig 2.10 shows frames from a video recording of a typical experiment using our soft robotic fingers to play the notes G, B, and D simultaneously (a G-major chord) under the control of the pneumatic RAM using TDM. In this experiment, the pneumatic RAM connected each finger to vacuum for 40 ms, cycling through all three fingers every 120 ms, or an update frequency of 8.33 Hz. At time $t = 0$, the pneumatic RAM has not yet begun operating and all four fingers are extended. After 7.2 s of operating the pneumatic RAM under TDM, Fingers 1, 2, and 4 are visibly contracting. By $t = 12.6$ s, all three fingers are fully contracted and the G-major chord is audible.

Ideally, in these experiments, all three notes in the G-major chord should sound simultaneously. In practice, at the fastest update frequency of 8.33 Hz, we observed up to 5 s of delay between the start of the first note in the chord and the start of the last note (Fig 2.10). This suggests that at the fastest update frequencies, the pneumatic RAM is simply running too fast for reliable operation. However, at update frequencies of 5.56 Hz and lower, the delay between the first and third notes in the chord drops to approximately 1 to 2 seconds. While still not simultaneous, the fingers are nonetheless contracting together in a reasonable amount of time, and this timing should be adequate for activities like grasping and releasing an object.

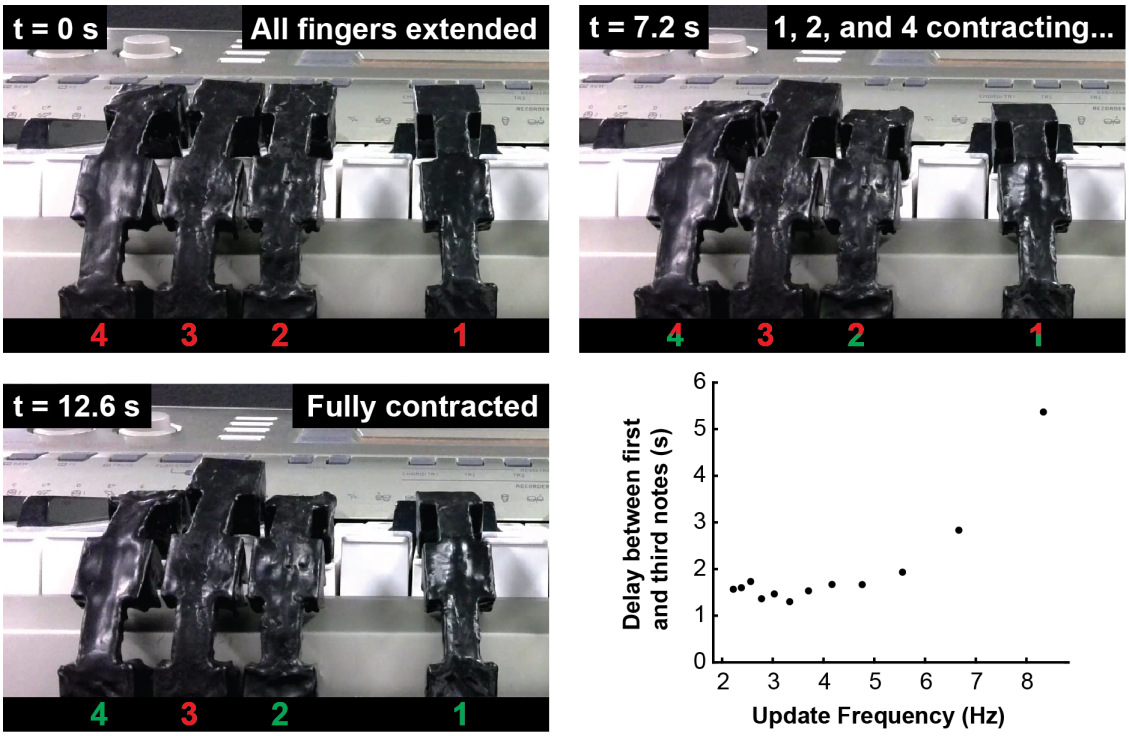

Figure 2.10: Video frames from using the pneumatic RAM and principles of time-division multiplexing (TDM) to control soft robotic fingers simultaneously playing the notes G, B, and D (a G-major chord) on an electric piano keyboard. The plot shows the observed delay between the start of the first note of the chord and the last note of the chord, as a function of the cycle speed of the pneumatic RAM. At cycle frequencies above 5.56 Hz the pneumatic RAM's performance degrades and the chord's notes do not sound simultaneously, but below 5.56 Hz the notes sound within 1 to 2 seconds of each other.

2.3.9 Playing a song on the piano

Finally, to demonstrate that the pneumatic RAM can control fingers both individually and simultaneously in a complex actuation pattern, we programmed the chip to play a song on the electric piano keyboard, in a manner akin to past demonstrations of piano-playing soft robots [57]. We wrote the arrangement of "Mary Had a Little Lamb" shown in the musical score in Fig 2.11 and programmed the pneumatic RAM to play the song. A video recording of the performance (available as online *Supplementary Information*) shows

that the soft robotic fingers played the song successfully and no errors were observed.

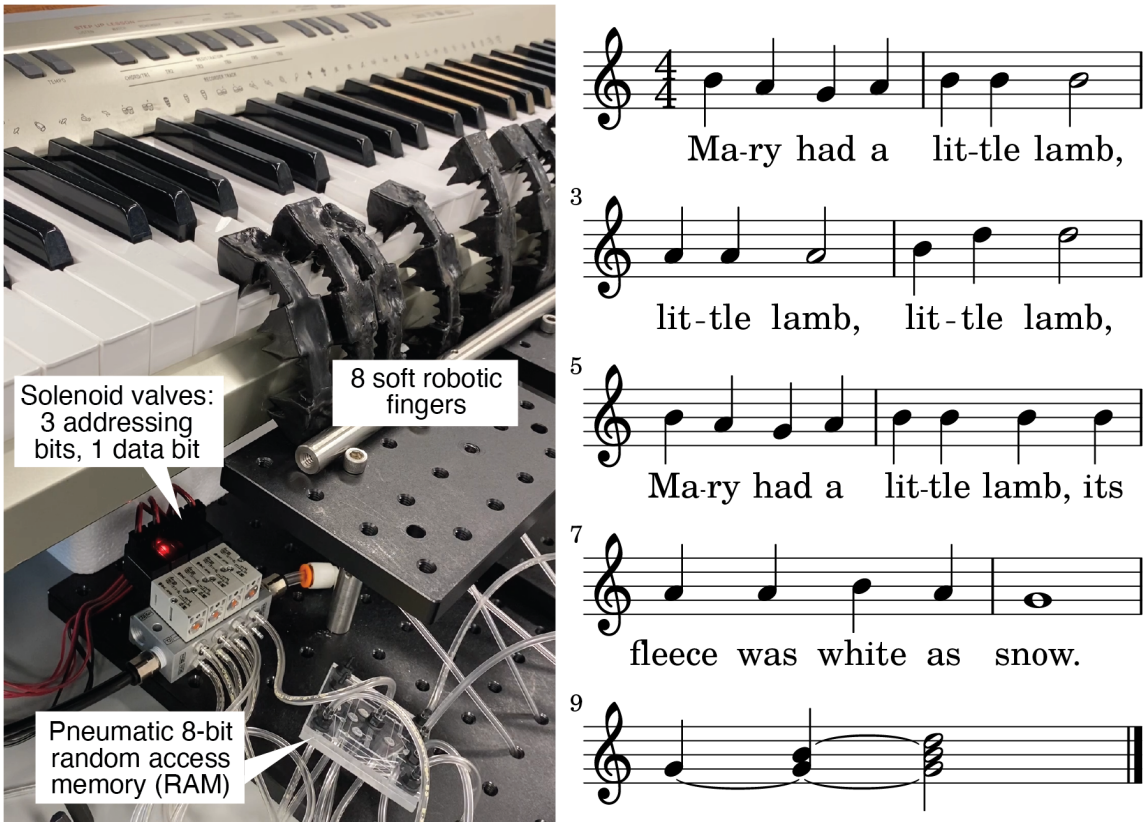

Figure 2.11: Video frame from using an eight-bit pneumatic RAM to control soft robotic fingers playing the music score shown on the right. The music consists of "Mary Had a Little Lamb" (measures 1 through 8; demonstrating playing one note at a time with varying durations) followed by an arpeggiated G-major chord (measure 9; demonstrating playing and holding multiple notes simultaneously). Source video available as online *Supplementary Information*.

2.4 Discussion

In this work, we introduced a pneumatic nonvolatile random-access memory (RAM) and showed that this pneumatic logic circuit can dramatically reduce the amount of peripheral hardware required to control a soft robot. To conclude, we examine the capabilities,

practical limitations, and future directions for this technology.

Based on our experimental findings, we noted earlier that a 10-bit pneumatic RAM could set 900 independent actuators to any desired actuation pattern. Would a pneumatic RAM chip this complex be feasible? To answer this, we can estimate some design parameters for this chip. In general, an n-bit pneumatic RAM capable of controlling 2^n independent soft actuators will contain $2^{n+1} - 2$ on-chip valves, so a 10-bit pneumatic RAM would contain 2046 on-chip monolithic membrane valves. With a surface area of about 7 mm^2 for each of our current valves, a 10-bit pneumatic RAM would contain about 140 cm^2 of valves; this is about the same surface area as a modern smartphone screen. While probably too large for some applications, a smartphone-sized pneumatic RAM would nonetheless be suitable for controlling many medium- and large-scale soft robots. And by using smaller monolithic membrane valves [18] or stacking several smaller pneumatic RAM chips, the overall footprint of the pneumatic RAM could be significantly reduced. Most importantly, this pneumatic RAM would require only 11 solenoid valves to operate it, a *99% reduction* in the amount of bulky, expensive, and power-consuming electromechanical hardware required to control a soft robot with 900 independent actuators. Since pneumatic soft robots usually require multiple solenoid valves (one per control line), a technology that exponentially reduces the number of solenoid valves required to operate a soft robot (while still maintaining complete individual control of each actuator) will result in significant reductions in the overall system cost, size, and power consumption of the soft robot.

For some applications, dramatically reducing the amount of electromechanical control hardware may not be enough—completely *eliminating* it would be the ultimate goal.

For example, soft robotic wearable devices can help infants with motor impairments lead normal lives [42], but the computers, microcontrollers, solenoid valves, batteries, and other hardware used to control these devices are difficult to safely use in close proximity to small children. Could pneumatic logic chips replace all this hardware? Replacing traditional computers and microcontrollers would require pneumatic logic circuits that can run complex multi-step programs, and these circuits have already been demonstrated at the microfluidic scale [34, 19, 59, 18, 33, 36, 35, 39, 46]. In principle, by replacing the monolithic membrane valves in these pneumatic computers with the high-flow valves we introduced in Fig 2.2C, one could make programmable pneumatic logic chips that control soft robots without the need for any electronic computing hardware. To power this pneumatic computer, we would still need a source of vacuum. This could be provided by a battery-powered pump, or for some applications a small evacuated tank could provide all of the power needed to power and control a soft robot for a period of time.

In its current form, the pneumatic RAM chip stores vacuums (pressures lower than atmospheric pressure). This means that the pneumatic RAM is limited to controlling vacuum-operated actuators like the soft robotic fingers used here. Vacuum-operated actuators are naturally limited to one atmosphere of pressure differential, and this makes them potentially "weaker" than pressure-operated actuators (which can in principle be powered by any desired pressure differential). For many applications that value the inherent safety and gentleness of soft robots, this limitation may actually be desirable: with no positive pressure anywhere in the system, there is no risk of accidental overpressurization and violent failure of the robot or its control system. However, to support the full range of pneumatic

actuators, a pneumatic RAM chip that can also store *pressures* would be necessary. Monolithic membrane valve-based pneumatic logic circuits capable of storing both vacuums *and* pressures in microfluidic applications have been demonstrated previously [24]; by implementing those circuits using the high-flow monolithic membrane valves described here, a pneumatic RAM capable of controlling *any* pneumatic actuator could be possible.

Finally, pneumatic RAM chips are not limited to controlling soft robots. Many traditional "hard" robots are powered by pistons and other pneumatic actuators, as is other equipment in manufacturing, construction, agriculture, mining, and other industries. Pneumatic RAM could reduce or eliminate electromechanical control hardware in these industries as well. In addition to providing cost savings, pneumatic RAM would be particularly useful in hazardous environments like coal mines, grain silos, and chemical plants, where electronic hardware could spark and cause fires or explosions.

2.5 Materials and Methods

2.5.1 Designing and fabricating pneumatic RAM chips

The pneumatic RAM chip in Fig 2.5 was designed in Adobe Illustrator (file available as online *Supplementary Information*), exported as a PDF file, and engraved into two pieces of poly(methyl methacrylate) (PMMA or acrylic) using a desktop CNC milling machine (Bantam Tools; Peekskill, New York). Each acrylic piece was 6.35 cm wide, 5.08 cm deep, and 3 mm thick. The channels milled into each layer were 280 μm wide and 254 μm deep. In the design in Fig 2.5, the channels connecting the central "Data bit" to the eight "Memory bit" connections (and thus to the soft robotic actuators) all have two, three, or

five parallel channels to accommodate greater air flow through these lines and therefore faster operation of the soft robot's actuators. We varied the number of parallel channels from two to five as an experiment, but in principle a chip designer can use as many parallel channels as they wish, with more parallel channels providing higher throughput of air and faster operation of the connected soft robot. Multiple parallel channels are not necessary leading to the valves' Control channels because these channels carry only small volumes of air flowing to and from the valves' Control chambers (not the larger volumes of air flowing to and from the soft robot's actuators). Each high-flow monolithic membrane valve inside the chip had a circular Control chamber with a diameter of 3 mm and a depth of 254 μm. The ports for the single Data bit input, six Address bit inputs, and eight Memory bit outputs had a diameter of 4.0 mm before being tapped with 10-32 threads and fitted with tubing connectors.

After engraving their channel patterns, the two PMMA pieces were bonded together with a featureless sheet of polydimethylsiloxane (PDMS) silicone rubber to form the completed pneumatic RAM. The 254 μm thick PDMS sheet (HT-6240; Rodgers Corporation/Bisco Silicones, Carol Stream, IL) was treated for 45 s per side using a handheld corona treater (BD-20AC; Electro-Technic Products, Chicago, IL) before bonding; this treatment strengthens the bond between the PDMS sheet and the PMMA pieces [27]. The engraved PMMA pieces were cleaned with 99.9% isopropanol, dried, then submerged in a 1% (volume/volume) solution of 3-aminopropyltriethoxysilane (Sigma-Aldrich, St. Louis, MO) in purified water for 20 minutes; this treatment also strengthens the PDMS-PMMA bond [88]. After drying, the bonding surfaces of the PMMA pieces were corona treated for 45 seconds

before sandwiching the PDMS sheet between the two PMMA pieces. The resulting pneumatic RAM chip was gently clamped and left overnight for the bond to strengthen before removing the clamp and using the chip.

2.5.2 Designing and fabricating soft robotic fingers

Soft robotic fingers were designed in SolidWorks (file available as online *Supplementary Information*), exported as an STL file, prepared for 3D printing using Ultimaker Cura software (0.1 mm layer height and 100% infill density), and fabricated using a 3D printer (Ender 3; Creality 3D Technology Co., Shenzhen, China). The filament used by the printer was thermoplastic polyurethane or TPU (NinjaFlex; Fenner Inc., Manheim, PA), a flexible filament with a Shore hardness of 85A. The 3D printer used an extruder temperature of 230 °C and an unheated (room temperature) print bed. After printing, small leaks in the soft robotic fingers were sealed by dipping the fingers in a rubber coating (Plasti Dip International, Blaine, MN) diluted to 50% in toluene, and then leaving the fingers to dry for at least four hours before use.

2.5.3 Controlling the pneumatic RAM chip

As shown in Figs 2.4, 2.5, and 2.6, the eight-bit pneumatic RAM is controlled by four pneumatic inputs: one Data bit input that receives vacuum or atmospheric pressure for routing to the selected Memory bit, and three Address bit inputs that determine which of the eight Memory bits (and associated soft robotic fingers) receives vacuum or atmospheric pressure. These inputs are provided by four "2 position, 4 way, 4 ported" solenoid valves (VQD1151-5L; SMC Corporation of America; Noblesville, Indiana) connected to a manifold

(VV4QD15-04M5; SMC). Each valve has two inlets, one connected to house vacuum (–68 kPa) and one left open to the atmosphere (this is the source of atmospheric pressure in our pneumatic logic circuits). Each valve also has two outlets, one connected to an Address input on the pneumatic RAM, and the other connected to the corresponding negated Address (or ¬ Address) input, as shown in Fig 2.5. At rest (not energized), each solenoid valve applies atmospheric pressure to the pneumatic RAM's Address input and vacuum to the ¬ Address input. When energized, the solenoid valve swaps these outputs, connecting vacuum to the pneumatic RAM's Address input and atmospheric pressure to the ¬ Address input. In this manner, the four solenoid valves perform the functions within the orange dashed box in Fig 2.4.

The four solenoid valves were controlled by programs (available as online *Supporting Information*) written in *OCW*, a simple language we developed for controlling microfluidic valves [23]. Pneumatic connections between the four solenoid valves, the pneumatic RAM, and the soft robotic fingers were made using flexible laboratory tubing (1/16" ID, 1/8" OD) as shown in Fig 2.11.

2.5.4 Characterizing pneumatic "memory"

A camera was used to record videos of a soft robotic finger while the pneumatic RAM trapped a vacuum inside the finger. The side view of the finger (shown in Fig 2.9A) and two white marks added to the finger facilitated the measurement of the finger's deflection angle over time using a custom MATLAB script (provided as online *Supplementary Information*). The pneumatic RAM chip continued to contract and extend other fingers while the test finger was held contracted by the trapped vacuum "memory." This test was

repeated for a total of three trials with a different actuator each time (all results in online *Supplementary Information*); the trial shown in Fig 2.9B had the median trapped vacuum duration of the three trials.

2.5.5 Playing notes and chords on a piano keyboard

A camera was used to record videos of several soft robotic fingers while playing chords (Fig 2.10) and songs (Fig 2.11) on a piano keyboard. The sound recording accompanying the video was used to determine the note-to-note delay times in Fig 2.10. The video recording of the soft robotic fingers playing "Mary Had a Little Lamb" (Fig 2.11) is available as online *Supplementary Information*.

Supporting information

1. Design of the pneumatic RAM chip in Adobe Illustrator format
2. Video recording of a pneumatic RAM showing valve states during operation
3. Table showing the state of each valve in each step during the operation of the pneumatic RAM in Fig 2.6
4. Design of the 3D-printed soft robotic finger in SolidWorks format
5. Video containing detailed frames of all 256 states of the eight soft robotic fingers from Fig 2.8
6. Code for measuring finger deflection angles in pneumatic memory experiments in MATLAB

7. Closeup of the first few seconds of Fig 2.9

8. Additional results from pneumatic memory characterization experiments like the one in Fig 2.9

9. Video recording of “Mary Had a Little Lamb” performed by a soft robot controlled by a pneumatic RAM chip

10. Source code for the OCW programs written to control pneumatic integrated circuits

Chapter 3

Pneumatic logic circuits for error detection

Shane Hoang, Konstantinos Karydis, Philip Brisk, William H. Grover

3.1 Abstract

Pneumatic systems play an important role in supporting key applications within sectors of modern infrastructures, such as hospitals, transportation systems, and manufacturing plants. This reliance on pneumatics makes it crucial for these systems to function optimally and in the case of malfunction, immediately alert personnel to resolve the issue at hand. In this work, we present an air-powered error detection chip that uses parity bit technology to detect modes of failure within vacuum-powered systems without the use of an electronic sensor. This device employs a circuit of monolithic membrane valves, arrayed as a combination of XOR gates, that inputs the pressure signals used to control the pneumatic

system to compute pressure values at its outlet. By comparing these outputs to those of the input pressure values, the error detection chip senses whether there is a change in the incoming signal from vacuum to atmospheric pressure, indicating that a broken connection (i.e. a leak) has occurred within the pneumatic system. We showcase how this pneumatic error detection chip can monitor the vacuum pressure used to operate a custom-designed intermittent pneumatic compression (IPC) device and alert its users during cases of malfunction. These results demonstrate how fluctuations in air pressure can be converted into a sensor output through pneumatic logic, with its versatility allowing it to be extended to a much broader range of pneumatic control systems.

3.2 Introduction

Pneumatics are used to control a wide variety of mechanical and industrial systems. Many pneumatic-controlled systems find applications in healthcare, manufacturing, transportation, and other areas where failures can have very serious consequences. In these critical applications, it is desirable to have error-detection strategies that can detect failures in the pneumatic control system (for example, a leak) and take appropriate action (raise an alarm, shut down the system safely, etc.). Often these error-detection strategies employ sensors that monitor air pressure or flow rate at various points in a system and relay this information to a computer for analysis. However, this electronic monitoring hardware adds considerable complexity, size, and cost to a system. This approach is particularly problematic in soft robotic systems that use pneumatics to control air-filled actuators: each pneumatic control line requires its own dedicated sensor, and adding all this electronic hard-

ware to a soft robot defeats many of the advantages of soft robotics (simplicity, autonomy, low cost, biomimetic design, few or no electronic components, and so on).

In this work, we demonstrate a pneumatic logic circuit that can detect errors in pneumatic control systems without using sensors or other electronic hardware. We accomplished this using *monolithic membrane valves*, a microfluidic valving technology that was originally developed for controlling fluid flow in microfluidics [25] but was later used to create air-powered logic circuits for controlling microfluidic chips [34, 24, 19, 33, 36, 35, 39, 46, 59, 18] and recently applied to controlling soft robots [78]. While means of error detection have been implemented in digital microfluidics [28, 55, 82], these methods mainly rely on auxiliary electronic sensing components to detect faults in the microfluidic circuits, which would increase the cost of and add unncessary complexity to soft robotic systems. In contrast, our pneumatic error detection circuit uses a *parity bit* for error detection; this basic but effective error detection technique has been used in electronic computing since at least the early 1950s [49]). Despite having been researched extensively in optoelectronic systems [60, 63, 16, 26, 12, 68, 71], parity bit error detection has so far been relatively unexplored in the field of microfluidics and can provide sensing capabilities solely using pneumatic signals. The circuit uses a network of monolithic membrane valves to calculate the parity bit based on the values of several pneumatic control signals; if the state of just one of these control signals changes, then the calculated parity bit will also change. If the value of this calculated parity bit differs from the expected value at any point during the operation of the system, then an error has been detected and the circuit outputs a pneumatic signal that can be used to alert a user, shut down the system, or take other action. As a proof-of-concept,

we used our pneumatic Error Detection Chip to detect failures in the operation of a model air-powered Intermittent Pneumatic Compression (IPC) device, a medical device that prevents the formation of life-threatening blood clots in the wearer's legs [64, 56, 11]. When a leak occurs that would compromise the efficacy of the IPC device and possibly endanger the wearer, the pneumatic logic circuit detects this error and alerts the wearer of the device by blowing a whistle. This pneumatic Error Detection Chip is a simple and low-cost way to add error detection to soft robots and other pneumatic-controlled systems.

3.3 Results

An overview of using the pneumatic Error Detection Chip to detect problems with the operation of a pneumatic-controlled system is shown in Figure 3.1. In this example, an Intermittent Pneumatic Compression (IPC) therapeutic device at right contains three air-filled segments that compress the wearer's leg when vacuum is applied to each segment in turn in a rolling or peristaltic manner. Normally, a computer (left) controls the state of three Control Bits, one for each air-filled segment of the IPC device. A value of 1 or TRUE for a bit indicates that vacuum should be applied to the corresponding IPC segment, and 0 or FALSE indicates that atmospheric pressure should be applied. These three electronic signals are passed form the computer to a set of solenoid valves, which effectively convert the electronic signals to their pneumatic equivalents and pass these signals to the IPC device.

To add our pneumatic Error Detection Chip to the IPC control system shown in Figure 3.1, a few small changes are necessary:

- First, the computer program is altered to also calculate the parity bit that corresponds

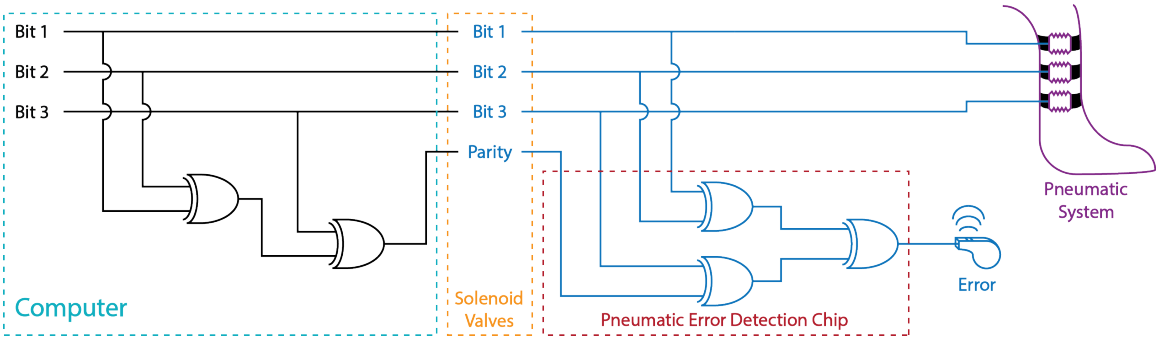

Figure 3.1: Overview of using the pneumatic Error Detection Chip

to the values of Control Bits 1, 2, and 3. This parity bit calculation can be thought of in a few different ways. One method calculates the Boolean Exclusive OR (or XOR) of the values of Control Bits 1 and 2, then calculates the XOR of the result and Control Bit 3; the result is the parity bit (this approach is diagrammed in Figure 3.1 at left). Alternatively, the sum Control Bit 1 + Control Bit 2 + Control Bit 3 (*modulo* 2) is also the parity bit. Finally, one can simply count the number of ones in the values of Control Bits 1, 2, and 3; if the count is odd then the parity bit is 1, and if the count is even then the parity bit is zero. All of these approaches are mathematically equivalent.

- Next, an additional solenoid valve is used to convert the parity bit calculated by the computer into its pneumatic representation (1 = vacuum, 0 = atmospheric pressure).

- Finally, the pneumatic Error Detection Chip is then connected to the three control bits and the parity bit. The connections to the control bits use "T" connectors so that the control signals still pass on to the IPC device. The pneumatic error detection chip

uses valve-based pneumatic logic gates (specifically, three pneumatic XOR gates) to calculate the parity bit based on the three values of the pneumatic control bits and compare this parity bit to the expected one calculated by the computer (details on chip operation below). If the two parity bits agree, then the values for control Bits 1, 2, and 3 have passed successfully from the computer to the IPC device; no error has occurred and the error detection chip outputs 0 or atmospheric pressure. However, if the two parity bits disagree, then something has happened that caused one of the control bits to have a different value at the IPC device than it did at the computer; an error has occurred. The error detection chip outputs 1 or vacuum; which in this example causes a whistle alarm to sound, alerting the wearer of the IPC device that a malfunction has occurred.

3.3.1 Error detection chip design and operation

The pneumatic error detection chip consists of three layers: a featureless polydimethylsiloxane (PDMS) membrane sandwiched between two engraved acrylic plastic sheets. Monolithic membrane valves [25] are formed wherever a gap in an engraved channel in one acrylic layer is located directly across the PDMS membrane from an engraved chamber in the other acrylic layer, as shown in the exploded view in Figure 3.2A. A cross-section through a valve (Figure 3.2B) shows that these valves are normally closed; the PDMS membrane normally rests against the channel gap and stops air from flowing across the gap. When a vacuum is applied to the chamber, the PDMS membrane is pulled into the chamber and away from the channel gap; this creates a path for air to flow across the gap and the valve opens. More generally, we can say that for a valve with pressures $P1$ and $P2$

at the two connections to the valved channel and pressure PC at the chamber:

- If $PC \geq P1$ and $PC \geq P2$, then the valve will be closed.
- If $PC < P1$ or $PC < P2$, then the valve will be open, and...
  - Air will flow from 1 to 2 as long as $P1 > P2$
  - Air will flow from 2 to 1 as long as $P2 > P1$

Multiple monolithic membrane valves can be connected together to form more complex pneumatic logic gates as shown in Figure 3.2C. For example, two valves in series function as a Boolean AND gate: the pneumatic signal at the input reaches the output only if Valve A AND Valve B both receive vacuum (that is, if A = TRUE and B = TRUE). Likewise, two valves in parallel function as a Boolean OR gate: the pneumatic signal at the input reaches the output if either Valve A OR Valve B (or both) receive vacuum.

A more complex logic gate that is integral to the operation of the pneumatic error detection chip is the exclusive-OR (XOR) shown in Figure 3.2C. This gate uses six valves, two vents (drilled holes to the atmosphere), one via (a hole punched through the PDMS membrane), and one Power input (a constant vacuum supply) to calculate the XOR of two bits A and B. If A = TRUE and B = FALSE, or if A = FALSE and B = TRUE, then the output of the XOR is TRUE (vacuum). However, if A and B are both TRUE, or A and B are both FALSE, then the output of the XOR is FALSE (atmosphere). A detailed explanation of the operation of this gate is shown in Figure 3.2C.

By connecting three XOR gates in the arrangement shown in the red box in Figure 3.1, we created the pneumatic error detection chip design shown in Figures 3.2D and 3.2E.

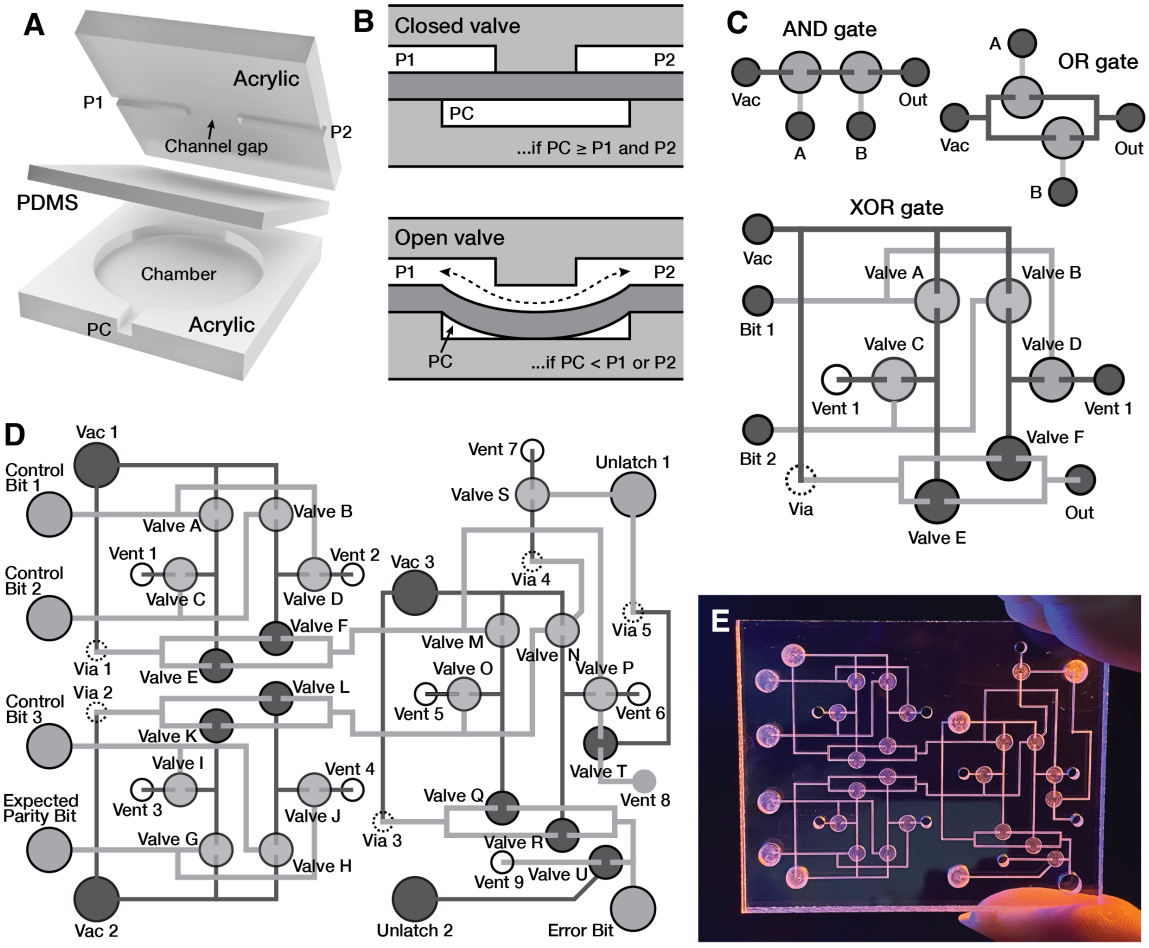

Figure 3.2: Exploded (A) and cross-section (B) views of a single monolithic membrane valve. Multiple valves can be connected by channels to make logic circuits like the Boolean AND, OR, and XOR gates (C). The Error Detection Chip consists of three XOR gates along with some additional support valves (D and E); it calculates the parity bit corresponding to three pneumatic Control Bit signals and compares the result to the input Expected Parity Bit.

This pneumatic circuit has 20 valves, four inputs (three Control bits and one Parity bit) and one output (the Error bit). The chip uses the XOR gates to calculate the parity bit corresponding to the values of the three Control bits, then compares this calculated value to the expected Parity bit value. When the two parity bit values agree, the Error output is at atmospheric pressure. However, if the two values for the parity bit differ, the chip outputs a TRUE signal (vacuum) on the Error output which can then be used to alert the user, shut down the system, and other functions.

Figure 3.3 shows the outcomes of calculations of the Error output, during operation of the pneumatic error detection chip, from three different states of Data bit configurations.

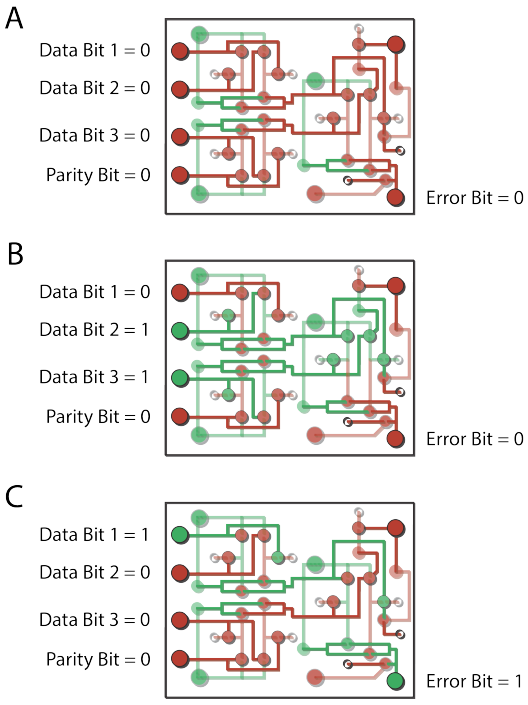

Figure 3.3: Three example calculations performed by the pneumatic Error Detection Chip. Channels containing vacuum are colored green, and channels containing atmospheric pressure are colored red.

3.3.2 Testing the pneumatic error detection chip

To test the operation of the pneumatic error detection chip, we applied all 16 possible combinations of three Control Bit and one Parity bit inputs while monitoring the Error output. Figure 3.4 plots the pressure at each of the four inputs and one output during a typical experiment (results from additional experiments are available as online *Supplementary Information*). On the left half of Figure 3.4, the input Parity bit is intentionally correct, and the pressure measured at the Error output remains at or close to atmospheric pressure (zero), correctly indicating that no error has occurred. However, in the right half of Figure 3.4, the input Parity bit is intentionally incorrect, and the pressure measured at the Error output always goes to vacuum, successfully indicating that errors have occurred.

3.3.3 Detecting errors in a model medical device

To validate the pneumatic error detection chip in a real-world application, we used it to monitor a model medical device, an Intermittent Pneumatic Compression or IPC device commonly used to prevent the formation of blood clots in the wearer's legs. Our model IPC device shown in Figure 3.5 consists of three flexible plastic bellows connected via 3D-printed buckles to nylon straps that wrap around a simulated leg. A computer program written in our valve control language *OCW* [23] applies vacuum to the three bellows one-at-a-time in a rolling or peristaltic pattern that is meant to encourage blood flow in the leg, and the same program also outputs the calculated parity bit corresponding to the states of the three bellows. As shown in Figure 3.1, the three pneumatic Control signals and one expected Parity bit signal are connected to the error detection chip, which repeats the

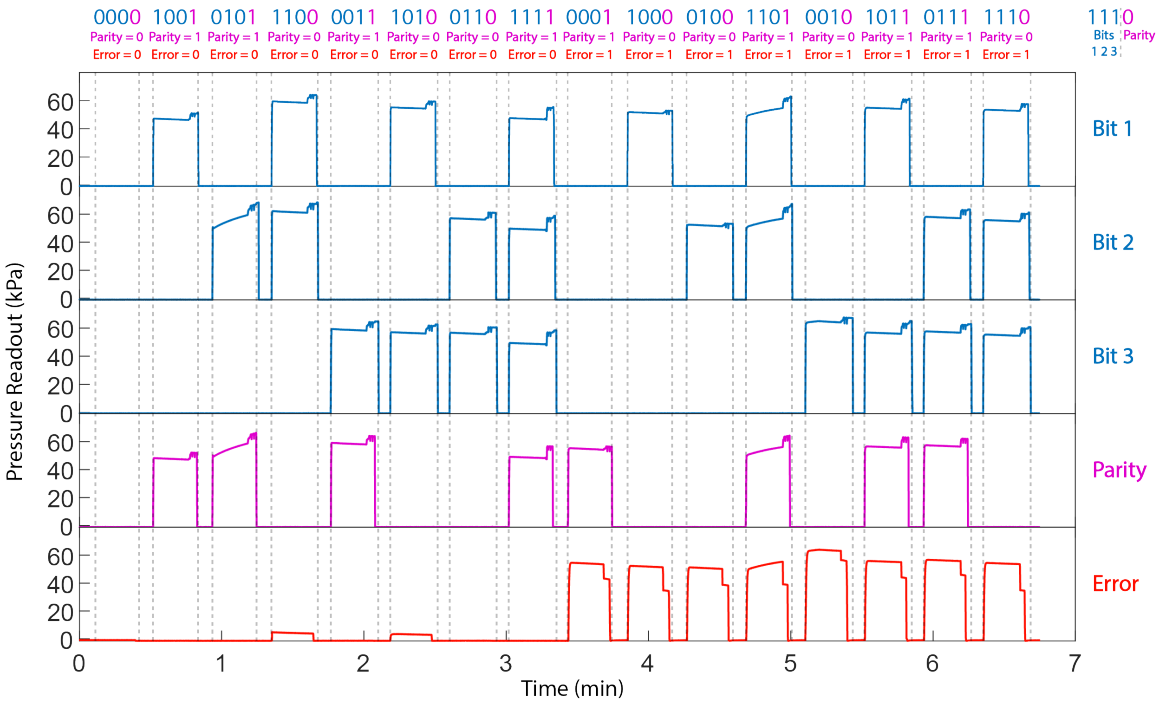

Figure 3.4: Pressure measurements at each of the three Control inputs, one Parity input, and one Error output while applying all possible combinations of inputs to the pneumatic error detection chip. During the first eight combinations (times from 0 to 3.5 minutes), the input Parity bit is correct or consistent with the values of the three Control bits, and the near-zero (atmospheric) pressure measured at the Error output confirms that no error has occurred. However, during the last eight combinations (times from 3.5 minutes to 7 minutes), the input Parity bit is intentionally incorrect (the opposite of what it should be), and the vacuum pressures measured at the Error output confirm that the chip has successfully detected these errors. Results from additional experiments like this are provided in online *Supplementary Information*.

parity bit calculation on the three Control signals and compares the result to the expected Parity bit. If the two values disagree, the chip applies a vacuum to the Error output.

In this demonstration, we wanted the error detection chip to alert the wearer by blowing a whistle when an error is detected. Since most whistles use positive pressure (not vacuum) to generate a sound, we needed a simple method for converting the vacuum at the Error output to a positive pressure for powering the whistle. We accomplished this using a"pneumatic level shifter" we developed as part of another project. This level shifter (shown in Figure 3.5) consists of a flexible plastic bellows mounted in a 3D-printed plastic frame in such a way that the bellows' motion is relayed to a pinch point in the frame containing tubing attached to a pressurized air supply. When no error is detected by the attached error detection chip, the level shifter's bellows is at atmospheric pressure and is fully extended, pressing against the pinch point and blocking the flow of pressurized air in the tubing. However, when an error is detected by the attached chip, the level shifter's bellows receives vacuum and contracts; this opens the pinch point and allows pressurized air to flow through the tubing and into the attached whistle, which makes a sound and alerts the wearer of a problem.

We successfully demonstrated two different modes of operation for the IPC device. In the first mode, the pneumatic Error Detection Chip is operated after each change in the Control Bits 1, 2, and 3. This mode offers continuous error checking (detecting an error as early as possible), but this comes at the expense of overall speed (operating the Error Detection Chip takes 45 seconds, so the Control Bits can only be updated every few seconds). A video recording of the IPC system in this mode of operation is available as

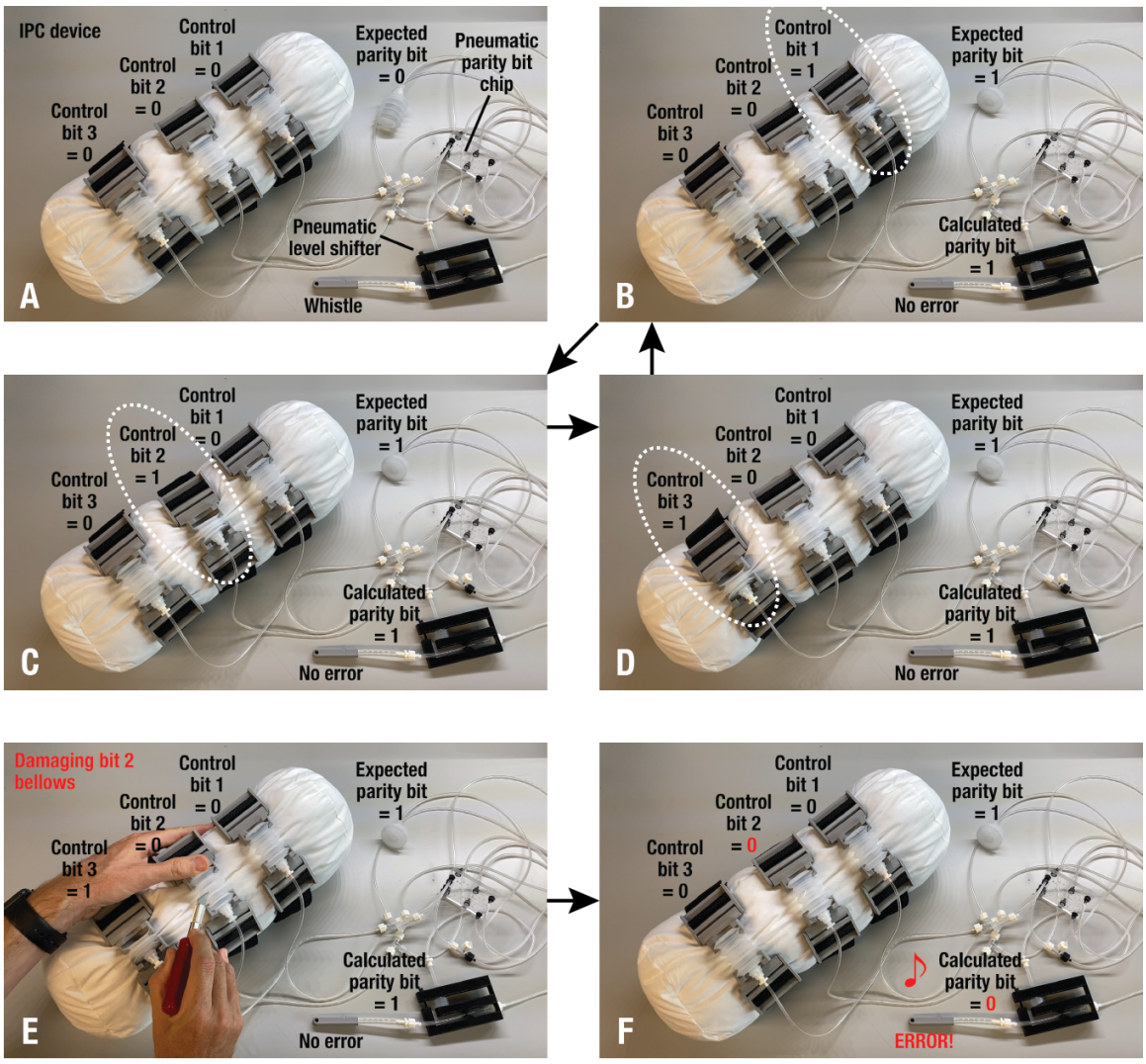

Figure 3.5: Frames from a video recording (available as online *Supporting Material*) of the pneumatic error detection chip connected to a model medical device, an Intermittent Pneumatic Compression or IPC device (A). During normal operation (the repeated cycle B to C to D), no errors are detected and the whistle connected to the error output is silent. When a bellows is intentionally cut to simulate an error (E), the error detection chip detects the leak as a mismatch in the expected and calculated values for the parity bit and automatically sounds the whistle (F).

online *Supporting Material*, and frames from this video are shown in Figure 3.5. The IPC device (Figure 3.5A) consists of three bellows connected to nylon straps wrapped around a soft simulated leg; the bellows contract and squeeze the leg when vacuum (1 or TRUE) is applied to the bellows via Control bits 1, 2, and 3, and extend or release the leg when atmospheric pressure (0 or FALSE) is applied via the Control bits. The pneumatic control hardware also calculates the expected parity bit corresponding to the values of the three Control bits and outputs this bit as a pneumatic signal (visualized using the fourth bellows in the upper-right). All four pneumatic signals are also connected to the pneumatic error detection chip, which repeats the parity bit calculation using the Control bits and compares the result to the value of the Expected parity bit to detect errors. During normal operation, the system repeatedly contracts one bellows at a time (dotted white ovals) to create a rolling or peristaltic squeezing motion (Figure 3.5B, C, D). With only one bellows contracted at a time, the number of 1s in the Control bits remains at one which is an odd number, so the Expected Parity Bit also remains at 1 (vacuum); the error detection chip verifies that the Expected and Calculated Parity Bits are the same and no error is detected. We then used a razor blade to damage the bellows on Control Bit 2 (Figure 3.5E). The next time that the system tries to apply vacuum to Control Bit 2 (Figure 3.5F), the air leak causes the Error Detection Chip to calculate a parity bit of 0 (Control Bits 1, 2, and 3 are all zero and 0 XOR 0 XOR 0 = 0), which disagrees with the Expected Parity Bit of 1. This causes the Error Detection Chip to output 1 (vacuum) on its Error output, which is in turn converted to a positive pressure using the black 3D-printed Pneumatic Level Shifter at the lower right, which is finally connected to a gray 3D-printed whistle. The positive air pressure blows the

whistle to alert the IPC wearer of the problem. The whistle continues to sound every time that the error is detected until the leak is repaired.

In the second mode of operation, the IPC system alternates between two phases: (1) rapid operation of the Control Bits without error checking for a period of time, and (2) checking each Control Bit using the Error Detection Chip for a period of time. This mode offers periods of much faster operation (the Control Bits can be updated several times per second during Phase 1) at the expense of error checking frequency (errors are only detected during Phase 2). In the video recording in *Supporting Material*, the IPC system alternated between spending 22.5 seconds(4.5 seconds per cycle, ran for 5 cycles) in Phase 1 (during which the Control Bits were updated every 0.75 seconds) and 39 seconds in Phase 2 (during which the system applied vacuum to the Control Bits one-at-a-time while checking for errors). When the IPC system was damaged during Phase 1 by using scissors to cut the tubing leading to Control Bit 3's bellows, the Error Detection Chip successfully detected this damage and blew the whistle 30 seconds later when the system entered Phase 2. Finally, this video also demonstrates that the Error Detection Chip's error signal is "reset" after fixing the error: when we repaired the cut tubing, the whistle was again silent in subsequent error-checking phases.

3.4 Discussion

In this work we introduced a pneumatic logic chip with parity bit functionality and showed how it could use pneumatic signals as a sensor for determining a failure in an IPC system in real-time. We conclude by discussing the implications, practical limitations,

and future directions for this technology.

The error detection chip provides a means of detecting failures in pneumatic systems through a layout of pneumatically-operated monolithic membrane valves. Through this design, the detection chip demonstrates the ability to sense drops in pressure in pneumatic chambers due to leaks, proving particularly beneficial to the field of soft robotics where many systems are centered around the manipulation of pneumatic chambers [97, 32, 98, 79, 96]. A key aspect of this system is the elimination of electronic sensors, made possible by using parity bit technology to detect when errors occur in pneumatic signals during transmission. Not only does this decrease the size and cost of soft robotic systems, but it also limits instances where electronic wiring can lead to short circuiting in the control hardware depending on the conditions the robot is placed in. The addition of the pneumatic level shifter, which acts to convert a vacuum pressure input into a high-pressure output, also opens many avenues for pneumatic logic. Despite its complexity as a pneumatic controller, vacuum logic is limited to the pressure present in the atmosphere; vacuum pressure is naturally unable to exceed -1 atm in relative pressure. However, the level shifter provides a means for using complex pneumatic logic to manipulate chambers that require much higher pressures to operate, allowing for the control hardware to scaled up to larger soft robotic systems.

As a by-product of using parity bit technology for detecting an error in pneumatic systems, the pneumatic error detection chip is limited in how many errors it can successfully detect. This is simply how parity bit logic works. Because parity bit checkers are looking at whether an incoming signal is distorted during transmission and use the binary scale,

multiple errors in the signal have the potential to cancel each other out and result in the detection of no errors. For example, looking at the 3-bit signal 1102 and calculating the expected parity bit, we would find that since there is an even number of 1's in the signal the expected parity bit would have a value of 0. As such, feeding this incoming signal and expected parity bit into our pneumatic error detection chip would yield an Error Bit value of 0, matching the expected value and showing that the signal has not been distorted. However, in the scenario where two errors occur simultaneously and changes the initial signal to 1012, the parity bit checker would not be able to detect the corrupted signal since there is still an even number of 1's in the signal, which by the rules of the parity bit would result in no error. Extending this further, having three errors occur in the transmission of this signal (i.e. 1102 changing to 0012) would trigger an error response since there would be an odd number of 1's between the altered signal and expected parity bit value; however, this would be detected as just one error. To summarize, having an odd number of errors occur would always result in a singular error while an even number of errors would be masked as there being no errors at all. While these limitations are inherently present in the parity bit layout, they only apply to scenarios where errors occur simultaneously. As a result, the pneumatic error detection chip is still able to function as intended for detecting singular errors and sounding off a whistle until the dysfunction is resolved.

In switching to a pneumatic version of a parity bit checker, which is made up of XOR gates, the error detection chip is subject to having pockets of latched vacuum form. Stemming from latching valves, these areas of trapped vacuum prevent valves from operating as intended and need to be reset after each set of valve operations for error detection. This

creates a delay in the operation of a pneumatic system, where a reset sequence needs to be implemented in addition to the series of valve operations needed to check the pressure of the pneumatic system for any failures. Despite this additional step, the reset sequence only requires five seconds to execute and should not significantly affect the performance of the error detection process. Furthermore, the two modes of operation of error detection provide users with an alternative to run this latch reset sequence after operation of the pneumatic system, during a subsequent error checking phase.

While this work focuses on the operation and error checking pertaining to IPC devices within the field of soft robotics, it should be reiterated that the pneumatic error detection chip is applicable to all systems that operate using a pneumatic chamber. As described earlier, pneumatics has played a key role in a variety of infrastructures and is heavily relied on for maintenance and automation. The versatility of this chip lies in its ability to be plugged into the middle of any of these pneumatic control systems and detect when failures occur simply from monitoring the air pressure already being used to control the system.

3.5 Materials and Methods

3.5.1 Error Detection Chip design and fabrication

The layout of the Error Detection Chip was modeled after a circuit of XOR gates, as shown in Figure 3.1, while being pneumatically powered. The chip was designed in Adobe Illustrator and exported as SVG files to Bantam Tools, where the features were milled onto two acrylic substrates (each 6.35 cm wide x 5.08 cm long x 3 mm thick) using a desktop

CNC mill (Bantam Tools; Peekskill, New York). Channels and membrane displacement chambers were engraved at a width and diameter of 284 μm and 3 mm, respectively, with both features having a depth of 254 μm. The vents and Error Bit were milled as through holes with diameters of 2 mm and 4 mm, respectively. Lastly, the ports for the Data Bits and Parity Bit were engraved to 4 mm in diameter and a depth of 2.75 mm. This leaves 0.25 mm of acrylic at the bottom of the port, where a pinhole is milled through with a diameter of 284 μm. Creating this pinhole allows for air to still pass through from the channels while also preventing the membrane from being pulled up into the port and potentially obstructing the path. Once the features were are all milled out, the ports were tapped with 10-32 threads.

After threading the ports, the two pieces of acrylic were cleaned using 99.5% isopropanol and soaked in a 5% (volume/volume) solution of 3-aminopropyltriethoxysilane (Sigma-Aldrich, St. Louis, MO), diluted in purified water, for 20 minutes. This acts to deposit alkoxysilane groups onto the surface of the acrylic substrates prior to bonding the pieces together. Following this surface modifcation, a 254 μm thick sheet of polydimethylsiloxane (PDMS) is cut out to the size of the acrylic pieces and punched with holes, using a 3 mm bioposy punch (Electron Microscopy Sciences; Hatfield, Pennsylvania), at the locations where channels on opposing layers of acrylic are joined together through vias. The acrylic and PDMS (HT-6240; Rodgers Corporation/Bisco Silicones, Carol, Stream, IL) were then treated with a handheld corona treater (BD-20AC; Electro-Technic Products, Chicago, IL) for 1 minute before bonding the pieces together, with the sheet of PDMS being sandwiched between the two acrylic substrates. The bonded chip was clamped overnight to allow the

bond to strengthen and then used the following day after removing the clamps and inserting tubing connectors to the ports.

3.5.2 Controlling the Error Detection Chip

The Error Detection Chip is controlled by three Data Bit inputs that carry the signal for operating the pneumatic system, a Parity Bit input that has been calculated from an XOR operation on the three bits of signal data, three Power Inputs for supplying constant vacuum to each of the three XOR gates, and two Latch Release Inputs for venting out areas of latched vacuum. These inputs receive either vacuum or atmospheric pressure, which are supplied by nine "2 way, 3 ported" solenoid valves (S070B-6BC, SMC Corporation of America; Noblesville, Indiana). Each valve has two inlets, with one connected to house vacuum (-68 kPa) and the other left open to atmosphere pressure, and an outlet that connects the solenoid valve to the corresponding Input. These solenoid valves supply atmospheric pressure to the Inputs at rest and supply vacuum pressure once energized.

During testing, the Error Detection Chip underwent 16 possible combinations of states for the bits to be in (8 where the parity bit was calculated correctly and 8 where the parity was calculated incorrectly), as shown in Figure 3.4. A custom multichannel pressure sensor circuit (MPX4250DP; NXP USA Inc.) and computer program was used to monitor the pressures at the Data Bits and Parity Bit (the printed circuit board design and Python code for the pressure monitor are available as *Supplementary Information*).

3.5.3 Operating the Error Detection Chip

As shown in Figure 3.1, the Error Detection Chip is made up of three XOR gates. Being that the system relies solely on pneumatics rather than electronics, these XOR gates are made up of a combination of pneumatic valves that provides the functionality of an XOR gate, as explained in Figure 3.2. Looking at Figure 3.2, it can be seen that each of the Data Bits as well as the Parity Bit are connected to a pair of valves in their respective XOR gates. When one of these inputs are supplied with vacuum pressure, the corresponding valves are pulled down and open. For example, if Data Bit 1 is supplied with vacuum pressure, then Valves A and D will be opened creating an open pathway for air to be pulled through. Conversely, supplying Data Bit 2 with atmospheric pressure will keep Valves B and C closed. As such, the states of each of the valves in these XOR gates will change depending on the signal being propagated from the computer to the pneumatic system.

Looking at the previous example with Data Bits 1 and 2, if the former is supplied with vacuum pressure and the latter with atmospheric pressure, then Valve A being open will allow Power Input 1 to pull down Valve E. Notably, this Power Input will also serve as the output of the XOR gate, due in part to Via 1 (a punched hole in the PDMS) connecting the two sides of the pneumatic chip. In the scenario where both Data Bits 1 and 2 are supplied with vacuum, then Valves A,B,C, and D will all be open. Consequently, this will cause Power Input 1 to pull air from Vent 1 rather than pull down Valve E, thus resulting in atmospheric pressure being the output of the XOR gate. This demonstrates the case where both inputs of the XOR gate are the same, preventing vacuum pressure from being the output of the gate.

In order to operate the XOR gates in the Error Detection Chip, the Power Inputs needs to be turned on to provide a constant supply of vacuum to the gate, as this will be the source of the output signal. Once all of the XOR gates have been powered by their respective Power Inputs, the combination of Data Bit values will determine the output of each gate, with the operation of the chip moving from left to right (from the Data Bits and Parity Bit towards the Error Bit). As shown in Figure 3.3, in scenarios where the sum of the input bit values is even (3.3A and 3.3B), there is no error detected, as indicated by the Error Bit having a value of 0. However, in the case where the sum of the input bit values is odd (Figure 3.3C), the necessary valves are opened in the chip to trigger the detection of an error (Error Bit having a value of 1), leading to users being alerted via a whistle sounding off. After alerting the presence of an error, the chip needs to be reset before detecting the next error; however, given the nature of Error Detection Chip's layout, as valves are opened and closed pockets of vacuum become trapped due to latching valves. These trapped areas of vacuum prevent the chip from operating normally and need to be cleared using a reset sequence with the two Latch Release inputs shown in Figure 3.2. The first step of resetting the Error Detection Chip is to close off the power to each of the XOR gates, working backwards from Power Input 3 (3, 2, and then 1). Once the power has been shut off, each of the Latch Release inputs can be turned on, which will open up Valves S and U to allow latched vacuum to exit out through Vents 7 and 9, and then turned off. Finally, the Data Bits needs to be reset to a value of 0 to close off all the valves on the left half of chip and prepare the Error Detection Chip for the next incoming signal.

3.5.4 IPC device design and fabrication

The IPC device consisted of three sets of 3D printed buckle constructs that were wrapped around a commercial pillow that acted as a model for a human leg. Each construct was made by using nylon straps to connect two 3D printed buckles that were designed with slots to house bellows (edges on the top and bottom of the bellows fit into grooves on the buckles). After inserting bellows into each buckle setup, each construct was placed around the pillow and fastened to create tight fits. Each bellows was then attached with tube fittings to create a pneumatic connection between each of the constructs and the data bits of the error detection chip. 3D printable files are available in *Supplementary Information*.

Supporting information

Files for *Supplementary Information* can be found in the online published version of this work.

Chapter 4

Pneumatic oscillator circuits for biomedical applications

Shane Hoang, Konstantinos Karydis, Philip Brisk, William H. Grover

4.1 Abstract

Many biomedical devices utilize periodic or oscillatory motions, and controlling these motions usually requires dedicated electromechanical hardware. This control hardware adds considerable expense and complexity to these devices and complicates their widespread use. In this work, we demonstrate a simple pneumatic oscillator circuit that can power oscillatory motions in biomedical devices using a single constant vacuum source and no electronics. This oscillator chip uses microfluidic monolithic membrane valves like transistors in air-powered logic circuits. A modification to the basic valve design—adding additional air channels in parallel through the valve—creates a "high-flow" valve that is

suitable for controlling larger volumes of air encountered in many biomedical devices. We designed a "high-flow" valve-based Boolean NOT gate, then arranged five NOT gates in a loop to create a self-oscillating pneumatic ring oscillator. The oscillator provides five out-of-phase pneumatic outputs that switch between vacuum and atmospheric pressure every 1.3 seconds. These outputs can be used to control a wide range of biomedical devices; in this proof-of-concept we used them to power a low-cost 3D-printed laboratory shaker or rocker commonly used to keep blood products, cell cultures, and other heterogenous samples in suspension. This work shows that many different biomedical devices can be made cheaper and safer using pneumatic logic circuits.

4.2 Introduction

Periodic or oscillatory motions play a key role in many biological and medical devices. For example, intermittent pneumatic compression (IPC) devices periodically squeeze a patient's legs to encourage blood flow and counteract the formation of clots, laboratory rockers and shakers use repetitive tilting or swaying motions to keep blood and cell cultures in suspension, and ventilators move air into and out of the lungs. Devices like these typically use electricity, motors or pumps, and computer or microcontrollers to create and control these periodic forces. All of this hardware adds considerable expense and complexity to the device. For example, while IPC stockings worn by patients are inexpensive enough to be single-use and disposable, the electromechanical hardware used to send periodic pneumatic signals to the stockings can cost thousands of dollars; this complicates the widespread use of IPCs in homes. Likewise, blood banks need large numbers of lab

rockers to keep blood products suspended and oxygenated and avoid coagulation; purchasing, powering, and maintaining all this electromechanical equipment can be a significant burden for health facilities in resource-limited settings. Additionally, electronic lab rockers and shakers may be unsuitable for use in some environments, such as high-humidity incubators (where moisture might encourage electrical shorts or corrosion), refrigerators and freezers (where condensation can damage electrical circuits), ovens (where overheating can damage motors and microprocessors), and flammable or oxygen-rich environments (where an electrical spark could cause a fire or explosion).

In this work, we describe a simple and robust pneumatic oscillator "circuit" that can power biological and medical devices like the ones described above without the need for electronic control hardware. The circuit uses air-powered microfluidic monolithic membrane valves that were originally developed for controlling liquids in microfluidic chips [25]. When multiple valves are connected together using air-filled microfluidic channels, they function like transistors in air-powered logic circuits [34, 24, 19, 33, 36, 35, 39, 46, 59, 18, 78]. Our pneumatic oscillator circuit is powered by a single constant vacuum source, which can be provided by house vacuum in a research or medical facility, or a simple low-cost vacuum pump for use at home or in resource-limited locations. The circuit converts this input vacuum into an arbitrary number of oscillating pneumatic outputs. These output signals oscillate out-of-phase with each other, which makes them ideal for controlling biomedical devices that utilize periodic or oscillatory motions. As a proof-of-concept, we used a pneumatic oscillator circuit to operate a 3D-printed laboratory rocker/shaker and confirmed that the device operates correctly and keeps blood samples in suspension. By dramatically

reducing or even eliminating the electronic hardware required to operate rockers/shakers and other biological and medical devices, pneumatic oscillator circuits can facilitate the widespread use of these important tools.

4.3 Results

Our pneumatic oscillator circuit uses monolithic membrane valves to create logic gates. As shown in Figure 4.1A, these valves consist of a featureless polydimethylsiloxane (PDMS) sheet sandwiched between two engraved acrylic plastic sheets. A valve is formed wherever an engraved chamber in one acrylic sheet is located directly across the PDMS membrane from a gap in a channel in the other acrylic sheet. Using multiple channels in parallel creates a "high-flow" valve suitable for containing larger air flows than are typically encountered in microfluidics [78].

The cross-sectional view through a single valve in Figure 4.1B shows that these valves are normally closed: when the same pressure is applied to the valved channels and the chamber, the PDMS membrane seals against the gap in the valved channels and no air flows through the valve. However, when a vacuum is applied to the chamber, the PDMS membrane stretches into the chamber and creates a path for air to flow across the gap in the valved channels (the dotted arrow in Figure 4.1), and the valve opens. More generally, for a valve with pressures $P1$ and $P2$ at the two ends of the valved channel and pressure PC at the chamber:

- If $PC \geq P1$ and $PC \geq P2$, then the valve will be closed.

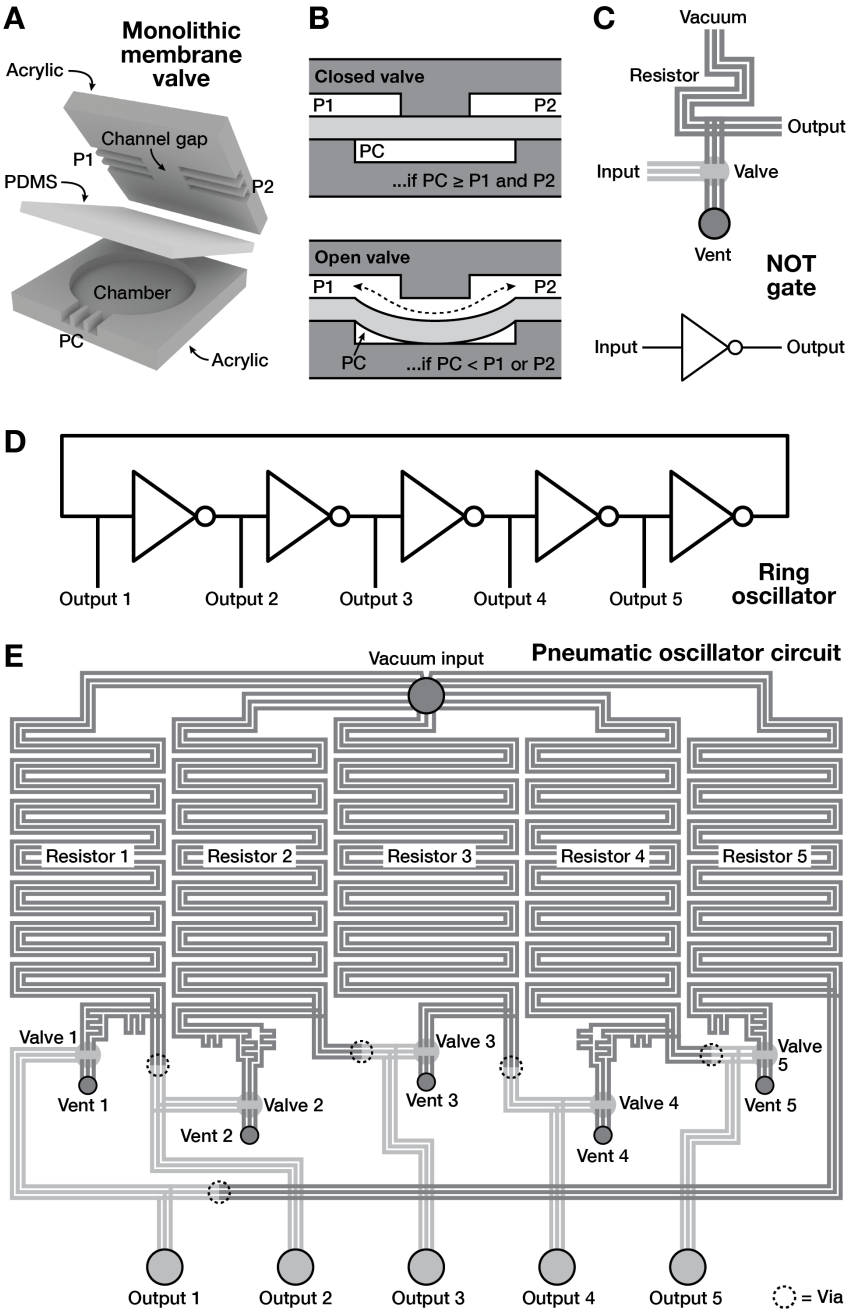

Figure 4.1: Exploded (A) and cross-section (B) views of a single high-flow monolithic membrane valve. Combining a valve with a vent hole and a long resistor channel creates a Boolean NOT gate (C) represented by the symbol shown. An odd number of NOT gates arranged in a circle creates a self oscillating circuit called a ring oscillator (D). Design of a pneumatic logic ring oscillator with five outputs (E). When a constant vacuum is applied to the vacuum input, five out-of-phase oscillating pneumatic signals are generated at the outputs.

- If $PC < P1$ or $PC < P2$, then the valve will be open, and...
  - Air will flow from 1 to 2 as long as $P1 > P2$
  - Air will flow from 2 to 1 as long as $P2 > P1$

Finally, by arbitrarily assigning a logical meaning of *TRUE* for a vacuum and *FALSE* for atmospheric pressure, binary information can be encoded and manipulated as different air pressure levels inside a microfluidic chip. In this manner, an infinite variety of pneumatic logic circuits can be constructed by connecting valves together using microfluidic channels.

The pneumatic oscillator circuit uses Boolean NOT gates; these fundamental logic gates output the opposite of their input (so if the input is *TRUE*, the output is *FALSE*, and if the input is *FALSE*, the output is *TRUE*). Figure 4.1C shows the design of the valve-based pneumatic NOT gate used in this study. The NOT gate is powered by a constant vacuum source that flows through a long section of channels that function as a pneumatic resistor. The output of the resistor splits; one end is connected to a valved channel, and other end is connected to the output of the gate. The other end of the valved channel is connected to a vent (a drilled hole that connects the contents of the channel to the atmosphere). Finally, the valve chamber is connected to the input of the gate.

When atmospheric pressure (the logical *FALSE*) is applied to the input of the pneumatic NOT gate in Figure 4.1C, the valve is closed. This means that air from the output can flow through the resistor to the vacuum source; this creates a vacuum (logical *TRUE*) at the output, as expected according the definition of the NOT gate. Conversely, when vacuum (logical *TRUE*) is applied to the input of the NOT gate, the valve is opened. This creates a low-resistance path from the output through the valve to the vent, effectively

putting the output at atmospheric pressure. While the vacuum source still pulls some air from the output, the different resistances of the two flow paths (the low-resistance path to the atmospheric vent vs. the high-resistance path to the vacuum source) ensure that the output is at atmospheric pressure (logical *TRUE*), again as expected for a NOT gate. In this manner, the pneumatic NOT gate aways outputs the opposite of its input.

When an odd number of NOT gates are connected in a loop as shown in Figure 4.1D, the resulting circuit is a ring oscillator [52]. This unstable circuit alternates the outputs between *TRUE* and *FALSE*. To understand why, imagine that the circuit starts with output 1 = *TRUE*. This is negated by the first NOT gate, so output 2 = *FALSE*, which makes output 3 = *TRUE*, which makes output 4 = *FALSE*, which makes output 5 = *TRUE*. This is negated by the final NOT gate to *TRUE*, which is then connected to output 1. This effectively flips output 1 from its original *FALSE* to *TRUE*, and it also causes all subsequent bits to flip. In this way, the values of all five bits automatically and constantly flip between *TRUE* and *FALSE*, with bit flipping propagating like a wave traveling around the loop.

By using five pneumatic NOT gates to build a ring oscillator, we created the pneumatic oscillator circuit design shown in Figure 4.1E. A single vacuum input at the top of the chip design powers all five NOT gates. Vias (holes punched through the PDMS membrane; dotted circles in Figure 4.1E) allow pneumatic signals to pass from one side of the membrane to the other. A photograph of a completed pneumatic oscillator chip is shown in Figure 4.2A.

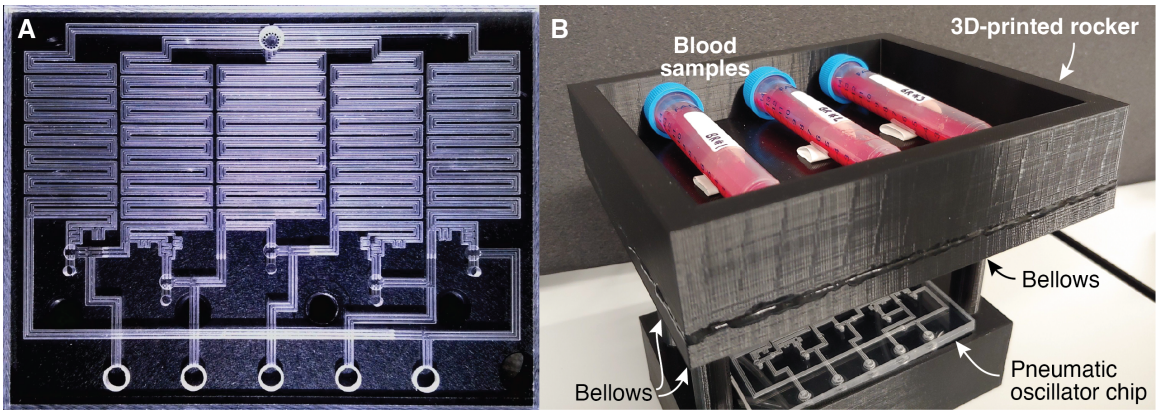

Figure 4.2: (A) Photograph of a completed pneumatic oscillator chip. (B) The oscillator chip installed in the base of a 3D-printed blood rocker. Three of the chip's five pneumatic outputs are connected to plastic bellows (not visible) that support the tray holding the blood samples. When a constant vacuum is applied to the chip, the oscillating output pressures the bellows to contract and expand one-at-a-time, creating a rocking motion that keeps the blood samples in suspension.

4.3.1 Characterization of Pneumatic Oscillator Circuit

To characterize the performance of the pneumatic oscillator chip, we connected its five pneumatic outputs to a custom-built open-source multichannel pressure logger (details in *Materials and Methods* below) and recorded the pressure in each output over time. Figure 4.3 shows typical results from operating the pneumatic oscillator nonstop for over two days. When viewing the entire dataset (Figure 4.3A), the individual oscillations in the five output pressures are not visible on this timescale, but we can see that the different outputs have different maximum vacuum pressures: Output 2 reaches the the highest vacuum at 35 kPa, Output 5 reaches 30 kPa, and the remaining three outputs reach maximum vacuums of between 15 and 20 kPa (this trend is explored further below). Though the maximum output vacuums are different for the different outputs, each output's maximum vacuum stays consistent over the two-day experiment.

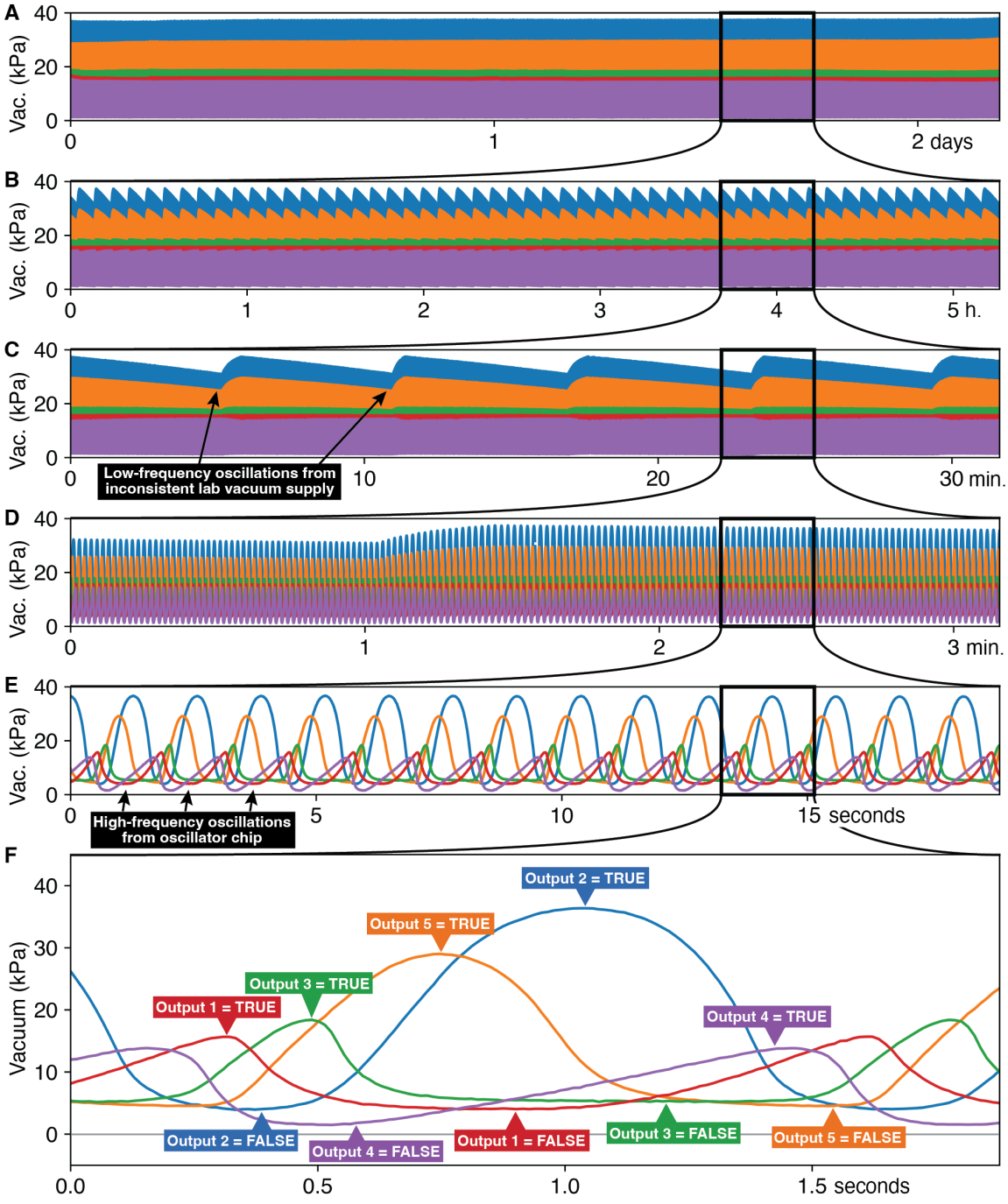

Figure 4.3: Vacuum pressure at each of the five oscillator chip outputs versus time during over two days of nonstop operation (A), and zooming in by successive factors of ten to view five hours (B), 30 minutes (C), 3 minutes (D), 15 seconds (E), and 1.5 seconds (F).

Zooming in on the time axis by a factor of ten yields the five-hour-long plot shown in Figure 4.3B. This plot reveals regular oscillations in the maximum vacuum recorded in the outputs. Zooming in by an additional factor of 10 (Figure 4.3C) shows that these oscillations occur about once every six minutes. These oscillations are far slower than the expected output of the pneumatic oscillator chip. We hypothesized that these oscillations could be caused by inconsistencies in the pressure of the vacuum used to power the oscillator chip. To test this hypothesis, we measured the pressure of our lab building's vacuum supply over time (see online *Supplementary Information* and found that the pressure of this vacuum supply varies with exactly the same timing and pattern as the output pressures in Figures 4.3B and C. Since we used this building vacuum supply to power the pneumatic oscillator chip, we attribute the low-frequency variations in Figures 4.3B and C to inconsistencies in the building vacuum supply pressure and note that these variations can be eliminated by using a more consistent vacuum supply. However, to ascertain the effect of inconsistent vacuum supplies on the operation of the pneumatic oscillator chip, we continued to use the building vacuum supply in the rest of this study.

After zooming in again by a factor of ten to view a 3-minute-long window in Figure 4.3D, the pressure changes produced by the oscillator chip become visible. On this scale, we can see that while the variations in the building vacuum supply do affect the maximum vacuum reached by the outputs (especially Outputs 2 and 5), the oscillator chip continues to operate as intended. Zooming in by another factor of ten (Figure 4.3(E) shows that the vacuum levels in the five outputs oscillate out-of-phase with each other, with each output reaching maximum vacuum at a different time; this is consistent with the expected operation

of a ring oscillator.

Finally, zooming in once more to view a window of 1.5 seconds (Figure 4.3F) reveals the pressure in each output during a single oscillation cycle. At this scale, we can trace changes in the pneumatic signals as they propagate around the ring. Starting arbitrarily on the left of Figure 4.3F with TRUE for Output 1, this signal is inverted to FALSE for Output 2, which is inverted to TRUE for Output 3, then FALSE for Output 4, and finally TRUE for Output 5; this portion of the oscillation sequence takes only about half a second to complete because the signal flows through adjacent NOT gates on the oscillator chip.

Here we discuss how each of outputs have different maximum vacuums, with each of those maxima lasting for different durations. This is useful, like having five signals to choose from, each with a different amplitude and duty cycle. During the first three hours, the average oscillation frequency was 0.749 Hz or an average period of 1.335 s per oscillation. In contrast, the last three hours had an average oscillation frequency of 0.779 Hz or 1.283 s/oscillation. Over the two-day-plus run, the oscillation period changed by (1.283-1.335)/1.335 = a 3.9% decrease in period. This decrease in frequency was due in part to the regulated laboratory source of vacuum pressure also displaying oscillatory outputs.

We also investigated the dependency of the oscillation frequency on the volume of the pneumatic chamber connected to the chip's outputs, namely the bellows, as shown in Figure 4.4. During this experiment, the bellows connected to Output 3 of the chip was compressed to observe how the frequency would change as a result. When the bellows was left at rest, the oscillation produced by the chip had a frequency of 0.798 Hz, or a period of 1.253 s, while compressing the bellows resulted in a frequency of 1.048 Hz, or a period of

0,954 s. As a result, compressing the bellows led to a 33% increase in speed , demonstrating how the volumes of pneumatic systems attached to the chip could affect the frequency of oscillation.

We then connected outputs 1, 2, 4, and 5 to a 3D-printed blood rocker shown in Figure 4.2B. The rocker was used to keep samples of blood in suspension, a common task for rockers in laboratories and blood banks. As confirmed in Figure 4.5, the oscillator-chip-powered rocker was able to successfully keep the blood samples in suspension, while a sample of blood left at rest displayed evidence of cells falling out of suspension.

4.4 Discussion

In this study we presented a pneumatic oscillator chip that is able to control multiple pneumatic systems in a continuous and oscillatory fashion, all while requiring only a source of vacuum pressure to operate. We conclude this work by discussing implications of this research, limitations of the system, and future directions for the technology.

By arraying together five NOT gates in series, the pneumatic oscillator circuit is able to produce a ringing oscillator effect, whereby the pneumatic output of the first logic gate is inverted and used as the input of the next. As the pneumatic signal propagates through the series of logic gates, the unsteady state of the system causes a ringing oscillation to occur. In electronics, this is achieved simply through a flip in a bit's data value; however, this is only possible in valve-based microfluidics due to the presence of vias that allow air flowing through the fluidic layer of the chip to cross over to the control layer and open up

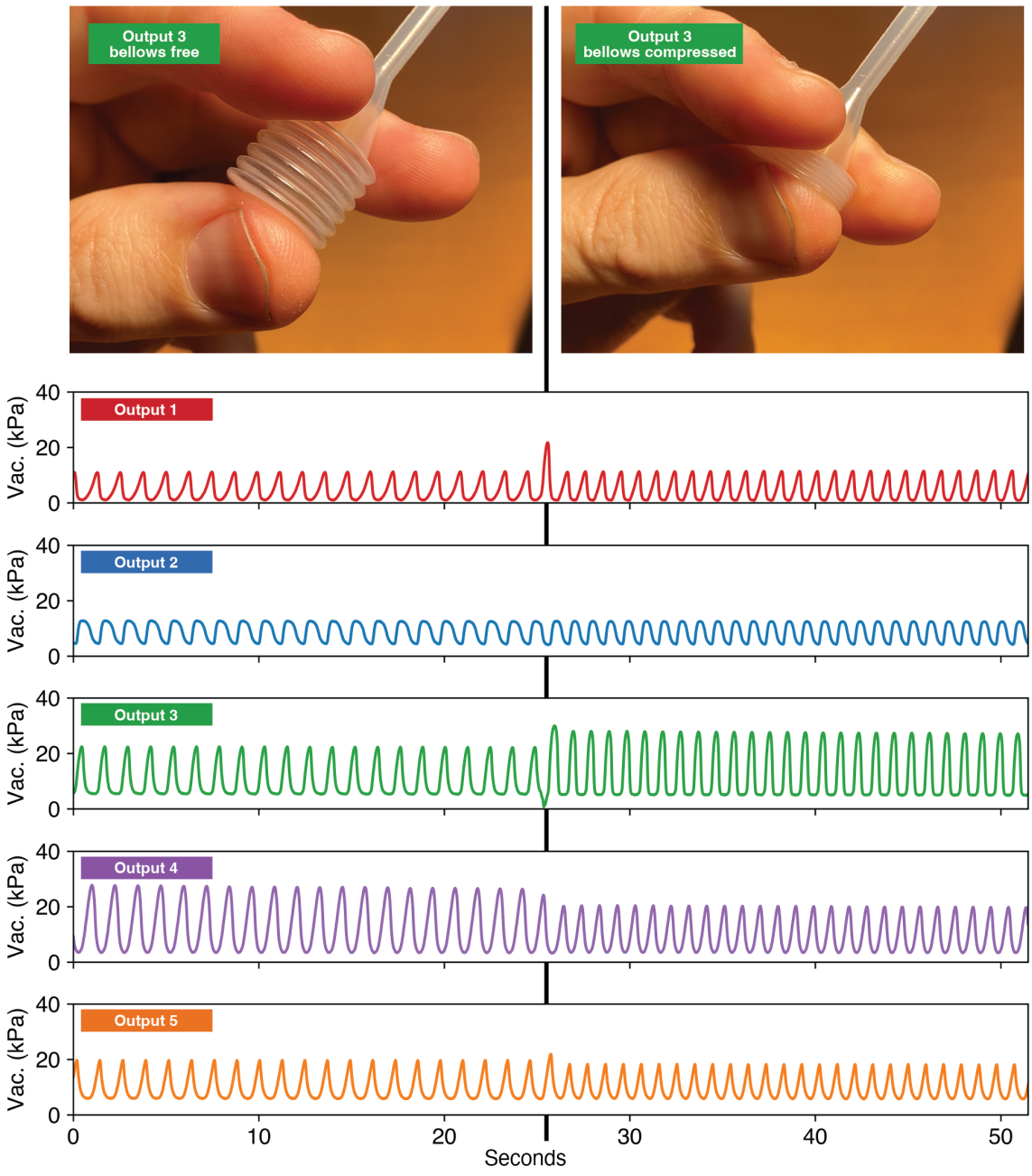

Figure 4.4: Vacuum pressure over time before and after compression of bellows at Output 3

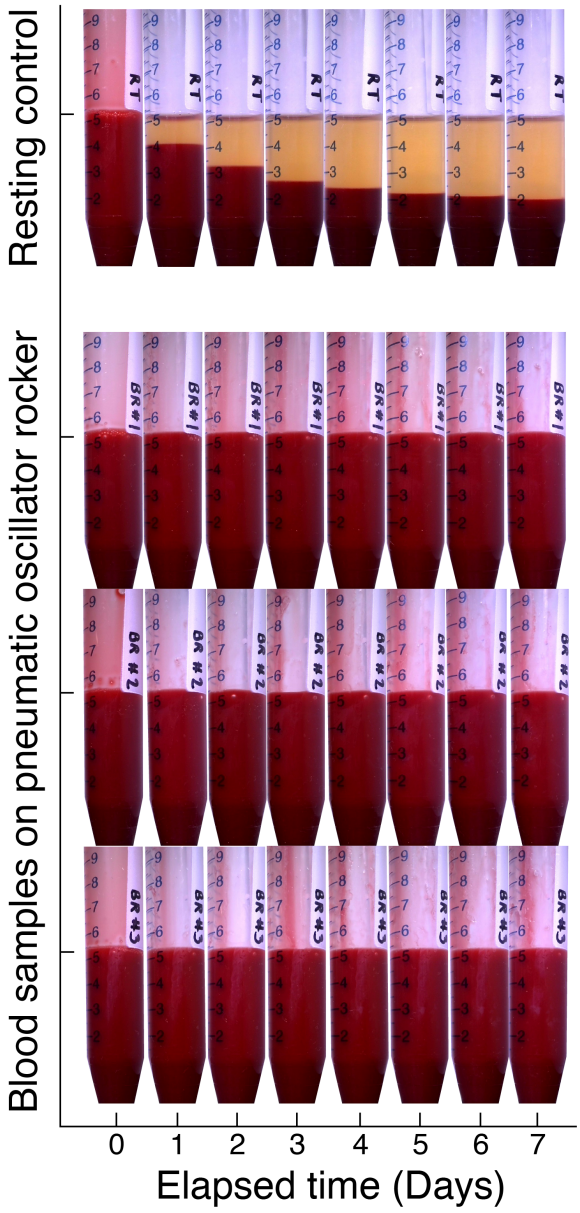

Figure 4.5: Suspension of blood samples over time

the next valve. While there would normally be concerns over the vacuum pressure being diminished after splitting to both open the valve and pass over the valve, the small size of the valves lends way to not requiring a large flow of vacuum pressure in order to operate. As such, the air flowing through the chip is able to effectively cause the valves to oscillate between states while also producing a pneumatic output substantial enough to control a soft robot at each of the chip's outlets. This presents a powerful approach to continuously operating pneumatic system without the need for any control hardware. As demonstrated in this work, systems like IPC devices would be able function with just a source of vacuum pressure to actuate each compression unit. Given that hospitals already allocate a pressure source to run these devices, the pneumatic oscillator chip could be seamlessly integrated into the system's workflow without any additional cost.

While the oscillator chip is able to function continuously, the frequency of oscillation could pose limitations on the device's applications. With the speed at which the valves oscillate inherently dependent on the microfluidic layout of the chip, the pneumatic oscillator circuit can only operate at speed. As a byproduct of eliminating all control hardware, the pneumatic outputs of the chip will oscillate at a set frequency as soon as the circuit is powered by a vacuum source, thus preventing a precise oscillation speed to be dialed in for a given pneumatic system. However, it has been demonstrated that manipulating the air within one of the outputs has a cascading effect that affects the speed of the ensuing oscillation. Because each of the outputs are connected to a pneumatic chamber, one or multiple of these chambers could be replaced with either a smaller or larger volume in order to zero in on a target oscillation frequency. Another limitation that this oscillator chip faces

is the potential buildup of dust over time. During operation of the chip, the valves are able to oscillate due to switching between a state where they are sequentially opened through vacuum pressure and another where atmospheric pressure is drawn in from vents to close them. Drawing in air has the potential to pull in dust to the chip and clog the channels, depending on the setting at which the chip is operating. While this could be remedied by adding a filter to each of these vents, there is also a risk that doing so will lead to a decrease in flow rate that would affect the effectiveness of the oscillation as a result.

Although the applications described in this study focused on operation through vacuum pressure, this does not limit the pneumatic oscillator chip from regulating systems that require positive pressure. All this would require is a conversion from the vacuum output of the oscillator device to a positive pressure input for the intended system. This is made possible through the concept of a level shifter, which is traditionally used to convert between voltage domains in digital electronics. This concept can be adapted for use in pneumatic logic as well since both vacuum and positive pressure operate on the same scale, just with different magnitudes of pressure. Future directions for this work also include the development of a model that can establish a relationship between the system's frequency of oscillation and the resistances within the chip due to channel dimensions and valve sizes. Precise tuning of oscillation frequency would allow for widespread application of this technology.

4.5 Materials and methods

4.5.1 Pneumatic Oscillator Chip design and fabrication

The pneumatic oscillator chip is designed to provide constant, pneumatic oscillatory outputs while operating with just a vacuum source, without the need for any control hardware (e.g. solenoid valves). This is accomplished by connecting five pneumatic inverters, or NOT gates, in series through a feedback loop. As shown in Figure 4.1, each NOT gate is comprised of a long resistor channel leading up to a valve and vent hole, with constant vacuum being applied to each through the vacuum input. All features of the chip were designed in Adobe Illustrator and then fabricated on two pieces of acrylic (each 6.35 cm in width, 5.08 cm in length, and 3 mm in height) with a desktop CNC mill (Bantam Tools; Peekskill, New York) using the Bantam Tools milling software. All channels were engraved to a width and depth of 450 μm while the membrane displacement chambers were milled out to a circular shape with a diameter of 3 mm and 450 μm deep to match the channel depth. The vents and outlets were milled out as through holes with diameters of 2 mm and 4mm, respectively. Lastly, the port for the vacuum input was engraved as a circle with a diameter of 4 mm and depth of 2.30 mm, leaving a 0.25 mm region of acrylic between the bottom of the port and the other side of the acrylic, as 0.45 mm is milled out on the other side for the channels. Pinholes, 0.45 mm in diameter, are then milled through this 0.25 mm region to connect the port to the channels on the opposite side. These pinholes act as through holes to allow air to be pulled through while also stopping the membrane from being drawn into the opening and causing an obstruction in the flow path. Once the inlet and outlets were all milled out, they were tapped with 10-32 threads.

Once all features of the acrylic have been milled out and threaded as necessary, the two pieces of acrylic were cleaned with 99.5% isopropyl alcohol and subsequently submerged into a 5% (v/v) solution of 3-aminopropyltriethoxysilane (Sigma-Aldrich, St. Louis, MO), diluted in purified water, for 20 minutes. This surface treatment deposits alkoxysilane groups onto the surface of the acrylic in preparation for bonding. Next, a sheet of polydimethylsiloxane (PDMS) is cut to the dimensions of the acrylic and punched with holes, using a 3 mm biopsy punch (Electron Microscopy Sciences, Hatfield, Pennsylvania), to form vias where channels from one side of the chip and connect to channels on the other side. The PDMS (HT-6240; Rodgers Corporation/Bisco Sciences, Carol, Stream, IL) and acrylic were then treated using a corona treater (BD-20AC; Electro-Technic Products, Chicago, IL) for 1 minutes before bonding the two pieces together, with the sheet of PDMS in-between. The chip was then clamped overnight to strengthen the bond and tested the following day after inserting tubing connectors to the inlet and outlets.

4.5.2 Controlling the Pneumatic Oscillator Chip

The oscillator chip is powered by an in-house laboratory vacuum connected the vacuum input on the chip. After propagating through the resistor channels, this constant vacuum input is able to supply a pneumatic signal to each respective outlet while also controlling the actuation of the valves, due to the presence of vias in each NOT gate. The pressure readings at these outlets were recorded using a custom Python script and multichannel pressure sensor circuit (MPX4250DP; NXP USA Inc.).

4.5.3 Blood Rocker design and fabrication

The blood rocker was designed to house tubes of blood in a compartment that is being shaken in a circular rocking motion due to the oscillatory output of the pneumatic oscillator chip. The structure of the rocker was designed in SOLIDWORKS and then fabricated with a commercial 3D printer (Ender-3, Creality) using PLA filament. The design allows for the oscillator chip to be housed below the rocker while also having four of the outputs be connected to bellows that rest at the four corners of the compartment holding the tubes of blood. The oscillatory vacuum output of the chip provides a rocking motion by compressing these bellows sequentially. An orifice was also designed at the side of the blood rocker to allow for variabilities in speed of the rocking operation by connecting different volumes of pneumatic chambers to it.

During testing, 5 mL of whole bovine blood (Lampire Biological Laboratories) was aliquoted into each of three 15 mL Falcon tubes that were placed side-by-side on the blood rocker. Vacuum pressure was then supplied to the pneumatic oscillator chip to power the rocker. The blood rocker was allowed to run for 7 days, with the Falcon tubes taken out (gently so not to disturb the suspension) each day to document signs of cells falling out of suspension. During this incubation, a fourth 15 mL Falcon tube, with an equivalent volume of whole bovine blood, was left upright at room temperature to act as a control for comparison. 3D printable files are available in *Supplementary Information*.

Supporting information

Files for *Supplementary Information* can be found in the online published version of this work.

Chapter 5

Conclusions

Still in its infant stages of development, the field of soft robotics has produced numerous soft robots with rubbery bodies that make them lighter and safer for human interaction, with the ongoing challenge being to replicate the functionality of traditional hard robots. Taking a biomimetic approach, researchers have been able to develop soft robots that mimic the functionalities of other living organisms, allowing them to grip objects, produce walking gaits and even traverse the ocean [10, 15, 20, 37, 45, 47, 53, 61, 67, 80, 83, 86]. However, performing these complex motions with fine precision still requires electromechanical components as control hardware. Motors and solenoid valves are still used to produce timely and consistent actuations in the robots, with both of these needing a power source to operate. The additional mechanical components and circuitry make these soft robots susceptible to malfunctions in poor weather conditions and can still be expensive to maintain in the long-term. As such, while many soft robots have become soft enough to pose minimal risks to other humans working alongside them, they still do not fully fit the image

of the ideal soft robot.

Given that the mechanical parts of soft robots are being used to control the movement of soft robots, researchers turned to other modes of producing actuation, leading to the use of combustion [8, 81, 90, 86]. Requiring only a fuel source, such as methane or butane, soft robots have demonstrated the ability to jump nearly a foot in height through a burst of movement produced from the large generation of pressure during the ignition of these combustible fuels. While completely free of electronics, this mode of locomotion is only able to produce vertical movement and also cannot execute precise controls due to the nature of combustion, thus exemplifying the need for functionality over control hardware reduction. While movement and actuation of soft robots can be achieved through chemical means without the need for electricity, if the result lacks the functionality required to mimic traditional robots, then the intended goal of soft robotics is still being missed. Although unable to fully eliminate all electromechanical hardware, pneumatic logic has proven capable of reducing the footprint of these components by a large margin while retaining precise motor capabilities of soft robots.

As demonstrated in this work, pneumatic logic is able to reduce the control hardware used to operate soft robots through the implementation of monolithic membrane valves. Powered by pneumatic means (vacuum pressure), these valves act as gates within microfluidic chips to limit the flow of air to connected robots and allow for soft robotic actuation at given times. By arraying these valves together, pneumatic logic gates can be formed in a circuit to allow for the control of multiple robots, impart memory by trapping vacuum pressure within the robot [78], produce continuous robotic locomotion, and even perform

error-checking during actuation. All that is required to produce these complex soft robotic functionalities is the designing of the logic circuit in accordance with the intended application, with even this step able to be expedited through computer science algorithms [54, 89] due to the monolithic nature of the circuit layout. However, the limitation of these applications is that they all still require a vacuum source to operate. After all, a constant source of manipulable vacuum pressure cannot be generated within these chips without external equipment. As such, these pneumatic networks focus on minimizing the footprint of additional mechanical components while imparting different modes of functionality in manipulating soft robotic actuators. Although relatively unexplored, the integration of this valve network into soft robots themselves has the potential for advancing the field of soft robotics towards full automation.

As soft robots have soft, rubbery bodies for safe interaction with humans, the main processes for fabricating the robots are either through 3D printing or molding silicone rubber. The method of 3D printing entails arranging melted plastic filaments in accordance with a designed file containing the surface geometries of an intended soft robot design. On the other hand, the silicone molding process uses molds (typically 3D printed but not required) as negatives, upon which commercial silicone rubber is poured and allowed to harden, resulting in the two halves of a soft robot, with empty spaces produced from the molds. By placing these halves together and sealing the resulting construct with silicone rubber, soft robots can be formed with a designed network of manipulable chamber space inside. While molding silicone rubber provides a simple and accessible way to produce soft robots, it is difficult to ensure the precision of features with fine details. In contrast, 3D

printing is slightly costlier due to requiring the machine but provides greater resolution of feature details, allowing for more intricate designs. Furthermore, the development of printers with multiple extruder heads allows for the interior of soft robots to have varying filaments to accommodate different functionalities and be fully printed within one operation. Using these properties, researchers have demonstrated the fabrication of a fully 3D printed robot with an embedded network of valves (normally-closed and normally-open) that is capable of locomotive actions as well as oscillatory actuation [29]. Although this robot requires a constant source of pressure to operate and uses control hardware to provide varying magnitudes of pressure, the embedded valve network represents a potential approach to fabricating soft robots with feedback loops. Soft robots that have been 3D printed with integrated valves could produce an oscillatory locomotion until a change in input pressure triggers a feedback loop to cause the robot to switch operation (e.g. different gait or switch directions). Such a robot would completely eliminate the need for solenoid valves and advance the field of soft robotics towards true mimicry of traditional robot functionality with minimal electromechanical components of control hardware.

While these limitations highlight a key aspect of soft robots in need of improvement, this does not take away from the rapid development that the field of soft robotics has experienced. Having being coined in 2008, the field of soft robotics has made leaps and bounds in the development of safer alternatives to rigid robots in just the span of a decade, during which biomimetic robotic functionalities have already been integrated into fabricated robots. Already capable of untethered, autonomous locomotion as well as sea explorations, soft robots have demonstrated immense potential for the future as viable and affordable

substitutes to the automated robots that have started to become prominent. Furthermore, the field of soft robotics presents itself as a very relatable area of research to many branches of science due to the theme of biomimicry, allowing researchers from varying backgrounds to develop new functionalities for soft robots through unique approaches.

During the time that soft robotics has been growing, a popular icon emerged as an optimal example of a soft robot: Baymax. While fictional, Baymax, a superhero character from the Disney animated film Big Hero 6, possesses many attributes that align with the goals of soft robotics. Namely, Baymax is a programmed robot with an inflatable soft exterior, capable of learning and safely interacting with humans, as well as provide medical diagnoses and proper treatment. However, even Baymax is not completely soft due to possessing a carbon fiber exoskeleton, showing that there is still room for improvement in the development of a truly 'soft' robot. Nonetheless, Baymax remains a popular figure in the field of soft robotics, with functionalities and qualities that have proven to be worthwhile in striving towards and ideally, even surpass.

Bibliography

[1] Heat-regulating apparatus, 7 1895.

[2] Adam R. Abate, Jeremy J. Agresti, and David A. Weitz. Microfluidic sorting with high-speed single-layer membrane valves. *Applied Physics Letters*, 96, 5 2010.

[3] Alfred V Aho and Jeffrey D Ullman. *Foundations of Computer Science*. Computer Science Press, Inc., 1992.

[4] Siavash Ahrar, Michelle Hwang, Philip N Duncan, and Elliot E Hui. Microfluidic serial dilution ladder. *Analyst*, 139:187–190, 1 2014.

[5] Alar Ainla, Mohit S Verma, Dian Yang, and George M Whitesides. Soft, rotating pneumatic actuator. *Soft Robotics*, 4:297–304, 5 2017. doi: 10.1089/soro.2017.0017.

[6] Hritwick Banerjee, Zion Tsz Ho Tse, and Hongliang Ren. Soft robotics with compliance and adaptation for biomedical applications and forthcoming challenges. *International Journal of Robotics and Automation*, 33:69–80, 2018.

[7] Nicholas W Bartlett, Kaitlyn P Becker, and Robert J Wood. A fluidic demultiplexer for controlling large arrays of soft actuators. *Soft Matter*, 16:5871–5877, 2020.

[8] Nicholas W Bartlett, Michael T Tolley, Johannes T B Overvelde, James C Weaver, Bobak Mosadegh, Katia Bertoldi, George M Whitesides, and Robert J Wood. A 3d-printed, functionally graded soft robot powered by combustion. *Science*, 349:161–165, 7 2015. doi: 10.1126/science.aab0129.

[9] Avraham Bromberg, Erik C Jensen, Jungkyu Kim, Yun Kyung Jung, and Richard A Mathies. Microfabricated linear hydrogel microarray for single-nucleotide polymorphism detection. *Anal Chem*, 84:963–970, 1 2012.

[10] M Calisti, G Picardi, and C Laschi. Fundamentals of soft robot locomotion. *Journal of the Royal Society Interface*, 14:0–2, 2017.

[11] A. H. Chen, S. G. Frangos, S. Kilaru, and B. E. Sumpio. Intermittent pneumatic compression devises - physiological mechanisms of action. *European Journal of Vascular and Endovascular Surgery*, 21:383–392, 2001.

[12] Kuladeep Roy Chowdhury, Debduti De, and Sourangshu Mukhopadhyay. Parity checking and generating circuit with nonlinear material in all-optical domain. *Chinese Physics Letters*, 22:1433, 2005.

[13] Theodore Christoforidis, Erik M Werner, Elliot E Hui, and David T Eddington. Vacuum pressure generation via microfabricated converging-diverging nozzles for operation of automated pneumatic logic. *Biomed Microdevices*, 18:74, 2016.

[14] Matteo Cianchetti, Cecilia Laschi, Arianna Menciassi, and Paolo Dario. Biomedical applications of soft robotics. *Nature Reviews Materials*, 3:143–153, 2018.

[15] Stephne Coyle, Carmel Majidi, Philip LeDuc, and K Jimmy Hsia. Bio-inspired soft robotics: Material selection, actuation, and design. *Extreme Mechanics Letters*, 22:51–59, 2018.

[16] E Dimitriadou, K E Zoiros, T Chattopadhyay, and J N Roy. Design of ultrafast all-optical 4-bit parity generator and checker using quantum-dot semiconductor optical amplifier-based mach-zehnder interferometer. *Journal of Computational Electronics*, 12:481–489, 2013.

[17] Dylan Drotman, Saurabh Jadhav, David Sharp, Christian Chan, and Michael T Tolley. Electronics-free pneumatic circuits for controlling soft-legged robots. *Science Robotics*, 6, 2021.

[18] Philip N Duncan, Siavash Ahrar, and Elliot E Hui. Scaling of pneumatic digital logic circuits. *Lab Chip*, 15:1360–1365, 3 2015.

[19] Philip N Duncan, Transon V Nguyen, and Elliot E Hui. Pneumatic oscillator circuits for timing and control of integrated microfluidics. *Proceedings of the National Academy of Sciences of the United States of America*, 110:18104–18109, 11 2013.

[20] Kevin C Galloway, Kaitlyn P Becker, Brennan Phillips, Jordan Kirby, Stephen Licht, Dan Tchernov, Robert J Wood, and David F Gruber. Soft robotic grippers for biological sampling on deep reefs. *Soft Robotics*, 3:23–33, 2016. PMID: 27625917.

[21] M Garrad, G Soter, A T Conn, H Hauser, and J Rossiter. A soft matter computer for soft robots. *Science Robotics*, 4, 2019.

[22] Benjamin Gorissen, Edoardo Milana, Arne Baeyens, Eva Broeders, Jeroen Christiaens, Klaas Collin, Dominiek Reynaerts, and Michael De Volder. Hardware sequencing of inflatable nonlinear actuators for autonomous soft robots. *Advanced Materials*, 31, 1 2019.

[23] William H Grover. Ocw: A really simple language for controlling microfluidic valves. https://github.com/groverlab/ocw.

[24] William H Grover, Robin H C Ivester, Erik C Jensen, and Richard A Mathies. Development and multiplexed control of latching pneumatic valves using microfluidic logical structures. *Lab on a Chip*, 6:623–631, 2006.

[25] William H Grover, Alison M Skelley, Chung N Liu, Eric T Lagally, and Richard A Mathies. Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices. *Sensors and Actuators B: Chemical*, 89:315–323, 2003.

[26] Bingchen Han, Junyu Xu, Pengfei Chen, Rongrong Guo, Yuanqi Gu, Yu Ning, and Yi Liu. All-optical non-inverted parity generator and checker based on semiconductor optical amplifiers. *Applied Sciences (Switzerland)*, 11:1–8, 2 2021.

[27] Kathryn Haubert, Tracy Drier, and David Beebe. Pdms bonding by means of a portable, low-cost corona system. *Lab Chip*, 6:1548–1549, 2006.

[28] Kai Hu, Bang Ning Hsu, Andrew Madison, Krishnendu Chakrabarty, and Richard Fair. Fault detection, real-time error recovery, and experimental demonstration for digital microfluidic biochips. pages 559–564. Institute of Electrical and Electronics Engineers Inc., 2013.

[29] Joshua D Hubbard, Ruben Acevedo, Kristen M Edwards, Abdullah T Alsharhan, Ziteng Wen, Jennifer Landry, Kejin Wang, Saul Schaffer, and Ryan D Sochol. Fully 3d-printed soft robots with integrated fluidic circuitry. *Science Advances*, 7:eabe5257, 11 2022. doi: 10.1126/sciadv.abe5257.

[30] Georges Ifrah. *The Universal History of Computing: From the Abacus to Quantum Computing*. John Wiley Sons, Inc., 2000.

[31] Filip Ilievski, Aaron D Mazzeo, Robert F Shepherd, Xin Chen, and George M Whitesides. Soft robotics for chemists. *Angewandte Chemie International Edition*, 50:1890–1895, 2011.

[32] Amir Janghorban and Reza Dehghani. Design and motion analysis of a bio-inspired soft robotic finger based on multi-sectional soft reinforced actuator.

[33] Erik C Jensen, Bharath P Bhat, and Richard A Mathies. A digital microfluidic platform for the automation of quantitative biomolecular assays. *Lab Chip*, 10:685–691, 3 2010.

[34] Erik C Jensen, William H Grover, and Richard A Mathies. Micropneumatic digital logic structures for integrated microdevice computation and control. *Journal of Microelectromechanical Systems*, 16:1378–1385, 2007.

[35] Erik C Jensen, Amanda M Stockton, Thomas N Chiesl, Jungkyu Kim, Abhisek Bera, and Richard A Mathies. Digitally programmable microfluidic automaton for multiscale combinatorial mixing and sample processing. *Lab Chip*, 13:288–296, 1 2013.

[36] Erik C Jensen, Yong Zeng, Jungkyu Kim, and Richard A Mathies. Microvalve enabled digital microfluidic systems for high performance biochemical and genetic analysis. *JALA*, 15:455–463, 12 2010.

[37] Robert K Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3:eaar3449, 3 2018.

[38] Jungkyu Kim, Erik C Jensen, Mischa Megens, Bernhard Boser, and Richard A Mathies. Integrated microfluidic bioprocessor for solid phase capture immunoassays. *Lab Chip*, 11:3106–3112, 9 2011.

[39] Jungkyu Kim, Erik C Jensen, Amanda M Stockton, and Richard A Mathies. Universal microfluidic automaton for autonomous sample processing: application to the mars organic analyzer. *Anal Chem*, 85:7682–7688, 8 2013.

[40] Jungkyu Kim, Minjee Kang, Erik C Jensen, and Richard A Mathies. Lifting gate polydimethylsiloxane microvalves and pumps for microfluidic control. *Anal Chem*, 84:2067–2071, 2 2012.

[41] Jungkyu Kim, Amanda M Stockton, Erik C Jensen, and Richard A Mathies. Pneumatically actuated microvalve circuits for programmable automation of chemical and biochemical analysis. *Lab Chip*, 16:812–819, 3 2016.

[42] Elena Kokkoni, Zhichao Liu, and Konstantinos Karydis. Development of a soft robotic wearable device to assist infant reaching. *Journal of Engineering and Science in Medical Diagnostics and Therapy*, 3, 2020.

[43] Tony Kuphaldt. Lessons in electric circuits, volume iv–digital, 2007.

[44] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics*, 1:1–12, 2016.

[45] Tiefeng Li, Guorui Li, Yiming Liang, Tingyu Cheng, Jing Dai, Xuxu Yang, Bangyuan Liu, Zedong Zeng, Zhilong Huang, Yingwu Luo, Tao Xie, and Wei Yang. Fast-moving soft electronic fish. *Science Advances*, 3:e1602045, 4 2017.

[46] Gregory Linshiz, Erik Jensen, Nina Stawski, Changhao Bi, Nick Elsbree, Hong Jiao, Jungkyu Kim, Richard Mathies, Jay D Keasling, and Nathan J Hillson. End-to-end automated microfluidic platform for synthetic biology: from design to functional analysis. *J Biol Eng*, 10:3, 2016.

[47] Zhichao Liu, Zhouyu Lu, and Konstantinos Karydis. A soft pneumatic hexapedal robot on rough, steep, and unstable terrain. pages 420–426, 2020. In Print.

[48] Xuanming Lu, Weiliang Xu, and Xiaoning Li. Pneunet based control system for soft robotic tongue. *2016 IEEE 14th International Workshop on Advanced Motion Control, AMC 2016*, pages 353–357, 2016.

[49] H Lukolf and H F Welsh. The uniservo-tape reader and recorder. page 47. IEEE Computer Society, 12 1952.

[50] Stephen T Mahon, Anthony Buchoux, Mohammed E Sayed, Lijun Teng, and Adam A Stokes. Soft robots for extreme environments: Removing electronic control. 2019.

[51] Carmel Majidi. Soft-matter engineering for soft robotics. *Advanced Materials Technologies*, 4:1–13, 2019.

[52] Mrinal Mandal and Bishnu Charan Sarkar. Ring oscillators: Characteristics and applications. *Indian Journal of Pure and Applied Physics*, 48:136–145, 12 2010.

[53] Andrew D Marchese, Cagdas D Onal, and Daniela Rus. Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. *Soft Robotics*, 1:75–87, 2014. PMID: 27625912.

[54] J McDaniel, W H Grover, and P Brisk. The case for semi-automated design of microfluidic very large scale integration (mvlsi) chips. pages 1793–1798, 2017.

[55] Debasis Mitra, Sarmishtha Ghoshal, Hafizur Rahaman, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. On-line error detection in digital microfluidic biochips. pages 332–337, 2012.

[56] R. J. Morris. Intermittent pneumatic compression - systems and applications, 5 2008.

[57] Bobak Mosadegh, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor J Walsh, and George M Whitesides. Pneumatic networks for soft robotics that actuate rapidly. *Advanced Functional Materials*, 24:2163–2170, 2014.

[58] M P Nemitz, C K Abrahamsson, L Wille, A A Stokes, D J Preston, and G M Whitesides. Soft non-volatile memory for non-electronic information storage in soft robots. pages 7–12, 5 2020.

[59] Transon V Nguyen, Philip N Duncan, Siavash Ahrar, and Elliot E Hui. Semi-autonomous liquid handling via on-chip pneumatic digital logic. *Lab Chip*, 12:3991–3994, 10 2012.

[60] Ali Norouzi and Saeed Rasouli Heikalabad. Design of reversible parity generator and checker for the implementation of nano-communication systems in quantum-dot cellular automata. *Photonic Network Communications*, 38:231–243, 2019.

[61] Preston Ohta, Luis Valle, Jonathan King, Kevin Low, Jaehyun Yi, Christopher G Atkeson, and Yong Lae Park. Design of a lightweight soft robotic arm using pneumatic artificial muscles and inflatable sleeves. *Soft Robotics*, 5:204–215, 10 2018. doi: 10.1089/soro.2017.0044.

[62] Arthur W J G Ord-Hume. *Pianola: the history of the self-playing piano*. George Allen Unwin, 1985.

[63] Nirmalya Pahari. All optical even and odd parity bit generator and checker with optical nonlinear material. *Journal of Optics*, 46:336–341, 2017.

[64] Hugo Partsch. Intermittent pneumatic compression in immobile patients, 2008.

[65] T Patino, R Mestre, and S Sánchez. Miniaturized soft bio-hybrid robotics: A step forward into healthcare applications. *Lab on a Chip*, 16:3626–3630, 2016.

[66] Panagiotis Polygerinos, Nikolaus Correll, Stephen A Morin, Bobak Mosadegh, Cagdas D Onal, Kirstin Petersen, Matteo Cianchetti, Michael T Tolley, and Robert F Shepherd. Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction. *Advanced Engineering Materials*, 19, 2017.

[67] Panagiotis Polygerinos, Zheng Wang, Kevin C Galloway, Robert J Wood, and Conor J Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. *Robotics and Autonomous Systems*, 73:135–143, 2015. Wearable Robotics.

[68] A J Poustie, K J Blow, A E Kelly, and R J Manning. All-optical parity checker with bit-differential delay, 1999.

[69] Daniel J Preston, Haihui Joy Jiang, Vanessa Sanchez, Philipp Rothemund, Jeff Rawson, Markus P Nemitz, Won-Kyu Lee, Zhigang Suo, Conor J Walsh, and George M Whitesides. A soft ring oscillator. *Science Robotics*, 4:eaaw5496, 6 2019.

[70] Daniel J Preston, Philipp Rothemund, Haihui Joy Jiang, Markus P Nemitz, Jeff Rawson, Zhigang Suo, and George M Whitesides. Digital logic for soft devices. *Proceedings of the National Academy of Sciences of the United States of America*, 116:7750–7759, 2019.

[71] Jayanta Kumar Rakshit, Jitendra Nath Roy, and Tanay Chattopadhyay. Design of micro-ring resonator based all-optical parity generator and checker circuit. *Optics Communications*, 303:30–37, 2013.

[72] Minsoung Rhee and Mark A Burns. Microfluidic pneumatic logic circuits and digital pneumatic microprocessors for integrated microfluidic systems. *Lab Chip*, 9:3131–3143, 2009.

[73] Steven I Rich, Robert J Wood, and Carmel Majidi. Untethered soft robotics. *Nature Electronics*, 1:102–112, 2018.

[74] Matthew A Robertson and Jamie Paik. New soft robots really suck: Vacuum-powered systems empower diverse capabilities. *Science Robotics*, 2:eaan6357, 8 2017.

[75] Philipp Rothemund, Alar Ainla, Lee Belding, Daniel J Preston, Sarah Kurihara, Zhigang Suo, and George M Whitesides. A soft, bistable valve for autonomous control of soft actuators. *Science Robotics*, 3:eaar7986, 3 2018.

[76] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robot using fluidic elastomer actuators. *Nature*, 521:467–475, 5 2015.

[77] Rob B N Scharff, Eugeni L Doubrovski, Wim A Poelman, Pieter P Jonker, Charlie C L Wang, and Jo M P Geraedts. *Towards behavior design of a 3D-printed soft robotic hand*, volume 17. 2017.

[78] Konstantinos A N D Brisk Philip A N D Grover William H Hoang Shane and Karydis. A pneumatic random-access memory for controlling soft robots. *PLOS ONE*, 16:1–25, 11 2021.

[79] Qi Shao, Xuguang Dong, Zhonghan Lin, Chao Tang, Hao Sun, Xin Jun Liu, and Huichan Zhao. Untethered robotic millipede driven by low-pressure microfluidic actuators for multi-terrain exploration. *IEEE Robotics and Automation Letters*, 2022.

[80] Robert F Shepherd, Filip Ilievski, Wonjae Choi, Stephen A Morin, Adam A Stokes, Aaron D Mazzeo, Xin Chen, Michael Wang, and George M Whitesides. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 108:20400–20403, 2011.

[81] Robert F Shepherd, Adam A Stokes, Jacob Freake, Jabulani Barber, Phillip W Snyder, Aaron D Mazzeo, Ludovico Cademartiri, Stephen A Morin, and George M Whitesides. Using explosions to power a soft robot. *Angewandte Chemie International Edition*, 52:2892–2896, 3 2013. https://doi.org/10.1002/anie.201209540.

[82] Vineeta Shukla, Fawnizu Azmadi Hussin, Nor Hisham Hamid, Noohul Basheer Zain Ali, and Krishnendu Chakrabarty. Offline error detection in meda-based digital microfluidic biochips using oscillation-based testing methodology. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 33:621–635, 10 2017.

[83] Sung Hyuk Song, Min Soo Kim, Hugo Rodrigue, Jang Yeob Lee, Jae Eul Shim, Min Cheol Kim, Won Shik Chu, and Sung Hoon Ahn. Turtle mimetic soft robot with two swimming gaits. *Bioinspiration and Biomimetics*, 11:36010, 5 2016.

[84] IEEE Staff. *2018 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[85] Seppe Terryn, Joost Brancart, Dirk Lefeber, Guy Van Assche, and Bram Vanderborght. Self-healing soft pneumatic robots. *Science Robotics*, 2:1–13, 2017.

[86] Michael T Tolley, Robert F Shepherd, Bobak Mosadegh, Kevin C Galloway, Michael Wehner, Michael Karpelson, Robert J Wood, and George M Whitesides. A resilient, untethered soft robot. *Soft Robotics*, 1:213–223, 2014.

[87] M A Unger, H P Chou, T Thorsen, A Scherer, and S R Quake. Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science*, 288:113–116, 4 2000.

[88] M E Vlachopoulou, A Tserepi, P Pavli, P Argitis, M Sanopoulou, and K Misiakos. A low temperature surface modification assisted method for bonding plastic substrates. *Journal of Micromechanics and Microengineering*, 19:15007, 11 2009.

[89] Junchao Wang, Naiyin Zhang, Jin Chen, Victor G.J. Rodgers, Philip Brisk, and William H. Grover. Finding the optimal design of a passive microfluidic mixer. *Lab on a Chip*, 19:3618–3627, 11 2019.

[90] Michael Wehner, Michael T. Tolley, Yiğit Mengüç, Yong Lae Park, Annan Mozeika, Ye Ding, Cagdas Onal, Robert F. Shepherd, George M. Whitesides, and Robert J. Wood. Pneumatic energy sources for autonomous and wearable soft robotics. *Soft Robotics*, 1:263–274, 12 2014.

[91] Michael Wehner, Ryan L Truby, Daniel J Fitzgerald, Bobak Mosadegh, George M Whitesides, Jennifer A Lewis, and Robert J Wood. An integrated design and fabrication strategy for entirely soft, autonomous robots. *Nature*, 536:451–455, 2016.

[92] George M Whitesides. Soft robotics. *Angewandte Chemie - International Edition*, 57:4258–4273, 2018.

[93] Amy Wu, Lisen Wang, Erik Jensen, Richard Mathies, and Bernhard Boser. Modular integration of electronics and microfluidic systems using flexible printed circuit boards. *Lab Chip*, 10:519–521, 2 2010.

[94] Ke Xu and Néstor O Pérez-Arancibia. Electronics-free pneumatic logic circuits for localized feedback control of multi-actuator soft robots. 2020.

[95] Dian Yang, Mohit S Verma, Ju Hee So, Bobak Mosadegh, Christoph Keplinger, Benjamin Lee, Fatemeh Khashai, Elton Lossner, Zhigang Suo, and George M Whitesides. Buckling pneumatic linear actuators inspired by muscle. *Advanced Materials Technologies*, 1:1600055, 2016.

[96] Te Faye Yap, Zhen Liu, Anoop Rajappan, Trevor J. Shimokusu, and Daniel J. Preston. Necrobotics: Biotic materials as ready-to-use actuators. *Advanced Science*, 10 2022.

[97] X Zhang, T Pan, H L Heung, P W Y Chiu, and Z Li. A biomimetic soft robot for inspecting pipeline with significant diameter variation. pages 7486–7491, 2018.

[98] Shumi Zhao, Yisong Lei, Ziwen Wang, Jie Zhang, Jianxun Liu, Pengfei Zheng, Zidan Gong, and Yue Sun. Biomimetic artificial joints based on multi-material pneumatic actuators developed for soft robotic finger application. *Micromachines*, 12, 12 2021.