

Bucknell University

Bucknell Digital Commons

Master's Theses

Student Theses

Summer 2023

Pneumatic Latched Demultiplexer Circuit for Controlling Multi-Actuator Soft Robots

Arsh Noor Amin
ana002@bucknell.edu

Follow this and additional works at: https://digitalcommons.bucknell.edu/masters_theses



Part of the [Electrical and Electronics Commons](#), and the [Other Mechanical Engineering Commons](#)

Recommended Citation

Amin, Arsh Noor, "Pneumatic Latched Demultiplexer Circuit for Controlling Multi-Actuator Soft Robots" (2023). *Master's Theses*. 269.

https://digitalcommons.bucknell.edu/masters_theses/269

This Masters Thesis is brought to you for free and open access by the Student Theses at Bucknell Digital Commons. It has been accepted for inclusion in Master's Theses by an authorized administrator of Bucknell Digital Commons. For more information, please contact dcadmin@bucknell.edu.

Pneumatic Latched Demultiplexer Circuit for Controlling Multi-Actuator Soft Robots

by

Arsh Noor Amin

A Thesis

Presented to the Faculty of

Bucknell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering

Approved:

DocuSigned by:



B72E134307FE4B0...

Adviser: Keith W. Buffinton

DocuSigned by:



60DA9A2AAA4443D...

Department Chairperson: M. Stu Thompson

DocuSigned by:



65E7F428C40C4C1...

Engineering Thesis Committee Member: Richard J. Kozick

DocuSigned by:



3A89CBF9EAED482...

Engineering Thesis Committee Member: Amal Kabalan

Date: 7/1/2023

Abstract

This thesis presents a novel method for controlling multi-actuator soft robots using pneumatic latched demultiplexer circuits implemented with monolithic membrane valves. The pneumatic circuits are designed to address the problems of heavy dependence on hard electrical components and lack of feasible multi-actuator soft robotic systems that limit the potential applications of soft robotics. The proposed pneumatic demultiplexers reduce the number of solenoid valves and electrical components required to drive soft robotic systems, making soft robot control mechanisms more compliant, scalable, and versatile. The thesis demonstrates the design, fabrication, and testing of 4 -bit and n -bit pneumatic demultiplexer circuits that can control 16 -actuator and 2^n -actuator soft robotic systems respectively. The thesis also evaluates the performance and functionality of the pneumatic circuits and discusses their potential applications in various fields of soft robotics.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Keith Buffinton, for his invaluable guidance, support, and encouragement throughout this thesis project. He has been a great mentor and a source of inspiration. I am also grateful to my thesis committee members, Prof. Amal Kabalan and Prof. Rich Kozick, for their constructive feedback, insightful suggestions, and helpful advice.

I would also like to thank Dan Johnson from the Mooney Lab for his generous assistance with the fabrication of the pneumatic circuits. His expertise and experience have been instrumental in the success of this project.

I am thankful to all my colleagues, professors, and friends at Bucknell for their friendship, collaboration, and support. They have created a stimulating and enjoyable research environment that has fostered my learning and growth.

Content

1. Introduction	1
1.1. Motivation	2
1.2. Problem Identification	7
1.3. Prior Work.....	8
1.4. Objective.....	12
2. Concepts	14
2.1. Electronics-Microfluidics Analogy	14
2.2. Monolithic Membrane Valves	17
2.2.1. P-MOS Transistor.....	18
2.2.2. D Latch	19
2.3. Demultiplexer	20
3. Methodologies	22
3.1. Design Overview	22
3.2. Controlling the Pneumatic Demultiplexer	26
3.2.1. Driver Hardware Setup	28
3.2.2. Generalized Controller Software	31

3.3. Microfabrication	34
3.3.1. Micromilling and Direct Bonding Method	35
3.3.2. Laser Engraving and Adhesive Method.....	40
3.4. Improvements	44
3.4.1. Channel Scaling	45
3.4.2. Branching	48
4. Evaluation and Demonstration.....	54
4.1. Latching Behavior	54
4.1.1. Latching Demonstration.....	55
4.1.2. Latching Period	58
4.2. Unlatching Behavior.....	66
4.2.1. Unlatching Demonstration	66
4.2.2. Frequency Response and Bandwidth	69
4.3. Rise Time	73
4.4. System Demonstration	76
5. Conclusion	79
5.1 Discussion	79

5.2 Future Work.....	80
References	83
Appendix.....	87
A. Controller Program	87
B. Laser Cutter Settings.....	94
C. Latching Period Plots	95
D. Rise Time Plots	101

List of Figures

Fig. 1. Conventional soft robotic system with three components: electrical control, solenoid valves, and actuators (left to right). 7

Fig. 2. Side view (left) and cross-sectional (right) view of the monolithic membrane valves, transistor-like elements for pneumatic digital circuits to illustrate their composition and working..... 18

Fig. 3. Monolithic membrane valve pneumatic inverter (left) and a p-MOSFET electrical inverter (right)..... 19

Fig. 4. Pneumatic demultiplexer with n input solenoid valves as select bits, input pressure as the data bit, and the 2^n actuators as outputs. 21

Fig. 5. Pneumatic AND gate with monolithic membrane valve (left). Equivalent electrical AND gate (with the same input and output labeling)..... 23

Fig. 6. 1-bit pneumatic demultiplexer circuit (left) and equivalent electrical demultiplexer circuit (right) developed using two AND gates and a negated input..... 25

Fig. 7. Huong et al.'s 3-bit pneumatic demultiplexer circuit (left) and equivalent electrical demultiplexer circuit (right)..... 26

Fig. 8. Pressure inputs needed to select the output O2 or 3rd output with a pressure response of -50 kPa using a 3-bit pneumatic demultiplexer (right). Digital select configuration to select output O2 or 3rd bit using a 3-bit electrical demultiplexer (left). 28

Fig. 9. Hardware setup needed to drive n-bit pneumatic demultiplexers (left). Hardware setup assembled to drive 3-bit and 4-bit pneumatic demultiplexers.....30

Fig. 10. Screenshot of the controller interface for driving and testing the pneumatic demultiplexers.....32

Fig. 11. Overview of the generalized nature of the controller software.33

Fig. 12. CAD drawings of the 3-bit pneumatic demultiplexer circuit developed for CNC milling the circuit.35

Fig. 13. Micromilling and direct bonding fabrication technique. (A) The first step involves micromilling channels and valves and tapping threaded holes onto PMMA. (B) The second step involves plasma treatment. (C) The third step involves sandwiching the PMMA-PDMS-PMMA layers.36

Fig. 14. 3-bit pneumatic demultiplexer chip fabricated using the micromilling and bonding technique.....37

Fig. 15. 2-dimensional vector graphic of the 3-bit pneumatic demultiplexer circuit developed for laser cutting.....40

Fig. 16. Laser engraving and adhesive fabrication technique. (A) The first step involves applying double-sided adhesive tape to each PMMA piece. (B) The second step involves laser engraving channels and valves and tapping threaded holes onto PMMA. (C) The third step involves removing the tape and sandwiching the PDMS to the PMMA with the adhesive.41

Fig. 17. 3-bit pneumatic demultiplexer chip fabricated using the laser engraving and adhesive method.....	42
Fig. 18. Pneumatic AND gates fabricated using laser engraving and adhesive method with varying channel depths of 2.7 mm, 1.5 mm, 0.8 mm, and 0.5 mm (from left to right).....	46
Fig. 19. Plot of volumetric flow rate against channel depth for pneumatic AND gates with varying channel depths illustrating the effect of channel depth on volumetric flow rate.	48
Fig. 20. 4-bit pneumatic demultiplexer designed using the branching technique. The red-bordered area shows the 3-bit pneumatic demultiplexer design used to develop the 4-bit version.	49
Fig. 21. 4-bit pneumatic demultiplexer chip fabricated using laser engraving and adhesive method.....	50
Fig. 22. Pressure against time plot illustrating the latching functionality of the 3-bit pneumatic demultiplexer where the circular markers represent the points at which the outputs were “turned off”	56
Fig. 23. Pressure against time plot illustrating the latching functionality of the 4-bit pneumatic demultiplexer where the circular markers represent the points at which the outputs were “turned off”	57
Fig. 24. Pressure against time plot to investigate decay type for long-term pressure decay in latched outputs for output O ₀ of the 3-bit pneumatic demultiplexer	59

Fig. 25. Pressure against time plot from Fig. 24 with a linear trendline and residual $R^2 = 0.994$60

Fig. 26. Pressure against time plot from Fig. 24 with an exponential trendline and residual $R^2 = 0.995$60

Fig. 27. Pressure against time plot from Fig. 24 with a 2-degree polynomial trendline and residual $R^2 = 0.997$ 61

Fig. 28. Pressure against time plot from Fig. 24 with a 3-degree polynomial trendline and residual $R^2 = 0.999$ 61

Fig. 29. Pressure against time plot to find the latching period for output O0 of the 4-bit pneumatic demultiplexer.63

Fig. 30. Pressure against time plot illustrating the unlatching functionality of the 3-bit pneumatic demultiplexer.68

Fig. 31. Pressure against time plot illustrating the frequency response of output O0 of the 3-bit pneumatic demultiplexer with a varying square wave as the input signal.....71

Fig. 32. Pressure against time plot illustrating the frequency response of output O7 of the 4-bit pneumatic demultiplexer with a varying square wave as the input signal. 72

Fig. 33. Pressure against time plot to find rise time for output O0 of the 3-bit pneumatic demultiplexer. 74

Fig. 34. LED demonstration for the pneumatic demultiplexer circuit showing pneumatic demultiplexer's binary states 00001000_2 , 10000011_2 , and 00101100_2 from top to bottom. 78

Fig. 35. Pressure against time plot to find the latching period for output O0 of the 3-bit pneumatic demultiplexer. 95

Fig. 36. Pressure against time plot to find the latching period for output O3 of the 3-bit pneumatic demultiplexer. 96

Fig. 37. Pressure against time plot to find the latching period for output O5 of the 3-bit pneumatic demultiplexer. 97

Fig. 38. Pressure against time plot to find the latching period for output O6 of the 4-bit pneumatic demultiplexer. 98

Fig. 39. Pressure against time plot to find the latching period for output O10 of the 4-bit pneumatic demultiplexer. 99

Fig. 40. Pressure against time plot to find the latching period for output O14 of the 4-bit pneumatic demultiplexer. 100

Fig. 41. Pressure against time plot to find rise time for output O3 of the 3-bit pneumatic demultiplexer. 101

Fig. 42. Pressure against time plot to find rise time for output O5 of the 3-bit pneumatic demultiplexer. 102

Fig. 43. Pressure against time plot to find rise time for output O6 of the 3-bit pneumatic demultiplexer. 103

Fig. 44. Pressure against time plot to find rise time for output O0 of the 4-bit pneumatic demultiplexer. 104

Fig. 45. Pressure against time plot to find rise time for output O6 of the 4-bit pneumatic demultiplexer. 105

Fig. 46. Pressure against time plot to find rise time for output O10 of the 4-bit pneumatic demultiplexer. 106

Fig. 47. Pressure against time plot to find rise time for output O14 of the 4-bit pneumatic demultiplexer. 107

List of Tables

Table 1. Theoretical truth table of the pneumatic AND gate representing the behavior of the pneumatic circuit with pressure values and the equivalent logical values, where “A” represents atmospheric pressure and “V” represents vacuum pressure.....24

Table 2. Components needed to assemble hardware setup to drive 3 and 4-bit pneumatic demultiplexers..... 31

Table 3. Protocol for the generalized controller software to establish communication between the microcontroller and host computer.34

Table 4. Experimental truth table for the functionality test of the 3-bit pneumatic demultiplexer fabricated using the micromilling and direct bonding technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned.....38

Table 5. Experimental truth table for the functionality test of the 3-bit pneumatic demultiplexer fabricated using the laser engraving and adhesive technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned..... 43

Table 6. Flow rates recorded for pneumatic AND gates manufactured using different laser power settings with increasing channel depths. 47

Table 9. Latching periods for 3-bit pneumatic demultiplexer.64

Table 10. Latching periods for 4-bit pneumatic demultiplexer..... 65

Table 11. Rise times for 3-bit pneumatic demultiplexer.....	75
Table 12. Rise times for 4-bit pneumatic demultiplexer.	76
Table 13. Laser cutter settings to fabricate pneumatic demultiplexer chips using laser engraving and adhesive method.	94

Chapter 1

1. Introduction

Robotics technology has advanced significantly in recent years, with a growing focus on the development of soft robots. The reason for the recent surge of interest in soft robotics can be attributed to the advantages soft robots have over traditional rigid robots. Rigid robots are typically manufactured using hard materials, such as metal or plastic, and are designed to perform specific tasks in a fixed environment. They are often used in manufacturing and assembly lines, where they are able to perform repetitive tasks with high precision and accuracy. However, rigid robots have limitations in terms of their ability to adapt to different environments and perform tasks that require flexibility and dexterity. Alternatively, soft robots are designed to be flexible and compliant, allowing them to adjust to different environments and perform tasks that require adaptability. They are made of materials such as silicone or rubber, which enable them to bend and twist in ways that rigid robots cannot.

Furthermore, the ability to deform allows soft robots to interact with humans and other objects in a comparatively gentler and safer manner. Soft robots can conform to their environment, making them less likely to cause serious harm. The materials used to manufacture soft robots are more adept at absorbing impacts and reducing transmission forces, minimizing the chance of injury or damage in the event of a

collision. In contrast, in the case of rigid robots, quantitative injury risk assessments have shown that traditional robots are heavy and hard, move rapidly, apply large forces, and typically handle hazardous tools which makes them dangerous to be in close proximity to humans (Matthias *et al.*, 2010).

In addition to their adaptability and safety, soft robots have several other advantages over traditional rigid robots. Soft robots are lightweight and easy to transport making them suitable for use in remote or inaccessible locations. Soft robots can be powered by a variety of energy sources, including electricity, pneumatic pressure, and chemical reactions, which makes them more versatile than traditional rigid robots.

1.1. Motivation

All the aforementioned features and advantages of soft robots allow this technology to have a wide range of potential applications in various fields. This provides a two-fold opportunity to bridge a gap that exists due to the limitations of rigid robots and to further develop a promising technology. For this thesis, some of the applications of soft robots were explored to understand the impact they create, the potential they hold, and the hindrances they face that limit their widespread adoption.

Researchers like Hawkes *et al.* (2017) have demonstrated the potential of soft robots in search-and-rescue, space exploration, and emergency response by developing soft robots that use growth as means of locomotion. Soft robots that replicate the locomotive behavior of fungal hyphae, neurons, and trailing plants can move through complex and unpredictable environments by lengthening along constrained paths and tight spaces

(Hawkes *et al.*, 2017). Hawkes *et al.*'s (2017) soft robots go beyond theoretical navigational devices as Blumenschein *et al.* (2018) have utilized this concept to manufacture tip-extending reconfigurable antennas. Blumenschein *et al.*'s antennas can be deployed in remote and unfavorable conditions and provide the ability to alter the fundamental operating characteristics of the antenna as needed (Blumenschein *et al.*, 2018). The practical features and functionalities of Blumenschein's antennas and Hawkes' navigating robots are possible solely due to the dynamic geometry of extending soft robots. Although growth-based soft robots as a concept hold a lot of potential, Hawkes and Blumenschein have only succeeded in developing elementary implementations of such robots. Hawkes' and Blumenschein's soft robots contain a single actuator controlled by a single pump which results in a short range and a decline in performance as the actuator extends. Expanding this implementation with multiple actuators and pressure sources would result in notable improvements in performance as more actuators can support extension in more directions and an increased number of sources would allow for more coverage and longer extensions. However, it would be difficult to incorporate such a multi-actuator system at a scale that is viable for the applications that Hawkes and Blumenschein have envisioned for their robots. This limitation is not uncommon in the field of soft robotics, as other researchers have struggled to increase the number of actuators in their soft robotics systems due to size constraints (Zaidi *et al.*, 2021). Hence, developing a viable method to increase the number of actuators in a soft robotic system is a major motivation for this thesis. Like Hawkes, Tolley *et al.* (2014) have developed navigational soft robots that are resilient to harsh conditions with similar applications like search-and-rescue and

exploration. However, unlike Hawkes *et al.*'s (2017) extending robots, Tolley's soft robots are quadrupedal, contain 6 independent Pneu-Net actuators, and utilize insect-like arachnid locomotion to navigate their environment (Tolley *et al.*, 2014). What makes Tolley's soft robots stand out from other multi-actuator navigational soft robots is the fact that they are untethered and can be driven for 2 hours while withstanding extreme temperatures and large external forces. Researchers have successfully tested these robots in environments like snowstorms, puddles of water, and direct flames with reported speeds of up to 18 m/hr. The reported specifications prove Tolley's soft robots to be incredibly resilient and suitable for a wide variety of applications. However, despite the impressive specifications and potential applications, Tolley's soft robots face the typical challenges of rigid robots, i.e., blunt impacts, applied pressures, and harsh environmental conditions due to the rigid control components that sit at the center of the soft robot's body. The rigid components include electrical components like air compressors, solenoid valves, and a microcontroller. Even though such components are common in typical multi-actuator soft robotic systems, usually "hard" control components are separated from the "soft" actuators to maintain the compliance property of the soft robot. However, in the case of Tolley's "untethered" soft robot such separation is not possible which is why Tolley's robots share disadvantages with their rigid counterparts. Reliance on hard electrical components to drive soft robots has been a challenge for many other similar soft robots that require some control components to be embedded into the soft robot's body (Iida & Laschi, 2011). Hence, developing a control mechanism that reduces the reliance on rigid electrical components is of immense importance.

In addition to exploration and emergency response, researchers have also demonstrated a great deal of interest in applying soft robotics to the field of medicine, especially minimally invasive surgeries. The aforementioned features of soft robots (safety, flexibility, and adaptability) allow for safe interaction with delicate parts of the human body with minimal risk of harming surrounding tissues and organs during surgery, making soft robots an ideal candidate for minimally invasive surgeries. An example worth mentioning is the soft robot surgical system developed by Wang *et al.* (2017) for cardiothoracic endoscopic surgeries. Wang *et al.* have successfully developed and tested a soft robotic surgical system that can be used to perform minimally invasive surgery on a beating heart (Wang *et al.*, 2017). Wang's system consists of a soft robot arm with three degrees of freedom controlled by cables that are attached to motorized pulleys. The system has been through rigorous preclinical tests with a 100% success rate. Despite positive preclinical test results, the diameter of the soft robot manipulator is too large for a minimally invasive surgery on a human. The large size of the manipulator can be attributed to the "feeding station" of the soft robotic system which contains servomotors, electrical wiring, microcontroller, and longitudinally driven cables. Wang's surgical system, like many other soft robotic systems, relies heavily on electrical control mechanisms, failing to capitalize on more compliant power sources like air. A pneumatic feeding station in this case could fulfill Wang *et al.*'s requirements for a suitable and more compliant soft robotic manipulator. A pneumatic feeding station does not need to be part of the manipulator, it can simply be connected to the manipulator using tubing, addressing the weight and size issues of the manipulator. Since pneumatic

soft robotics control mechanisms have applications in soft robotic surgical systems, investigating such control mechanisms is another motivation for this thesis.

Another medical application of soft robotics that is worth mentioning is the soft robotic pad called “IntelliPad” developed by Raeisinezhad *et al.* (2020). In their work, Raeisinezhad *et al.* aim to prevent pressure injuries like bedsores by introducing a soft robotic pad constructed of a grid of individual dome-shaped actuators made out of compliant materials (soft silicone rubber) and driven by pressurized air and vacuum (Raeisinezhad *et al.*, 2020). The dome actuators are individually driven in response to the skin-pad interaction forces at each actuator which allows IntelliPad to redistribute the applied loads. Raeisinezhad *et al.* performed multiple tests to evaluate the effectiveness of IntelliPad in preventing pressure injuries. The tests included experimentation with simulated weights that successfully demonstrate the potential of this concept. However, Raeisinezhad’s work lacks experimentation in real-world settings. Furthermore, the IntelliPad used for testing consisted of partial dome-actuator grids with at most 3 adjacent dome actuators. Although Raeisinezhad *et al.* aim to expand this implementation with a full grid IntelliPad with 81 dome actuators in the future, such a system will require a fairly complex control mechanism. Currently, each dome actuator used in the IntelliPad contains three separate air chambers, each of these actuators utilizes a solenoid valve as a driver. Hence, a full grid IntelliPad with 81 dome actuators will require 81 solenoid valves. Having separate solenoid valves for each actuator severely limits the implementation of IntelliPad as the increase in the number of actuators causes an exponential increase in the complexity of the system. This problem is common in soft robotic systems that require 50, 100, or even 1000s of

individual actuators (Iida & Laschi, 2011). Thus, investigating ways to reduce the number of solenoid valves needed for multi-actuator soft robots is another motivating factor for this thesis.

1.2. Problem Identification

Through the previous discussion, it is clear that soft robotics is a field worth exploring. The hurdles that limit the applications of soft robotics from being widely implemented, however, make for interesting problems. Two such challenges previously identified are heavy dependence on hard electrical components for soft robot control and lack of feasible multi-actuator soft robotic systems. These challenges become more obvious when a conventional soft robotic system like Tolley *et al.*'s (2014) resilient untethered soft robot is considered. Soft robots typically consist of three components: pneumatic actuators, solenoid valves, and electrical control (as shown in Fig. 1).

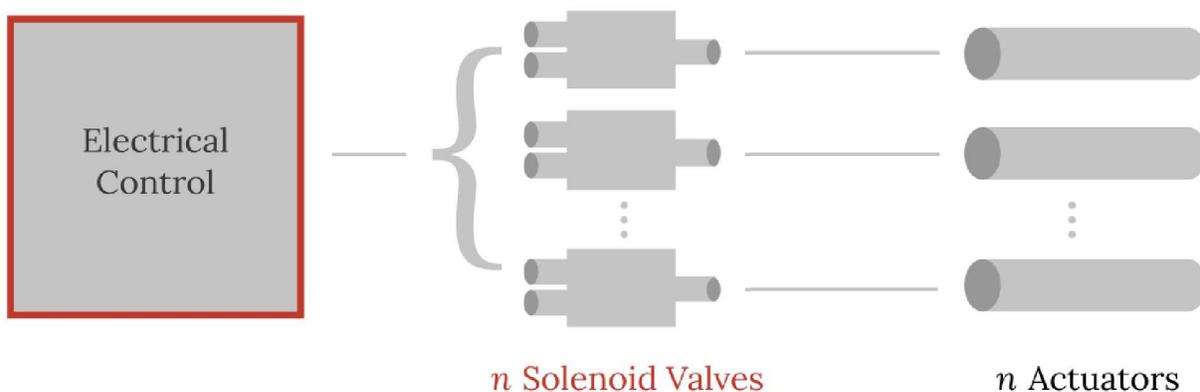


Fig. 1. Conventional soft robotic system with three components: electrical control, solenoid valves, and actuators (left to right).

Pneumatic actuators are driven by pressurized air or vacuum to expand or contract. This flow of air or vacuum is usually provided using electromechanical solenoid valves. Hence, each actuator, in a conventional soft robotics system, requires its own dedicated solenoid valve which prohibitively complicates multi-actuator soft robotics systems that require 100s or even 1000s of actuators like the IntelliPad (Raeisinezhad *et al.*, 2020). Moreover, to drive the solenoid valves electrical components like relays and microcontrollers are needed which further complicates the system. Hence, the need for large numbers of solenoid valves, along with the electrical control for those valves, makes current multi-actuator soft robotic systems bulky and expensive in terms of their cost and power requirements. Moreover, reliance on rigid electrical components as drivers of the soft robot makes them less compliant and hazardous, which limits their potential applications.

1.3. Prior Work

To address the problems described above in the section 1.2. Problem , researchers have endeavored to combine the electrical and pneumatic aspects of soft robot control systems in the form of pneumatic microfluidic logic. For reference, microfluidic devices allow the manipulation of fluids on a small scale (10^{-9} to 10^{-18} liters) using channels with dimensions in the 10s and 100s μm range (Whitesides, 2006). Recent advancements in the field of microfluidics have created avenues to address the problems with soft robot control mechanisms. For instance, through microfluidic lab-on-chip implementations, Zhang *et al.* (2017) have demonstrated how microfluidic solutions offer advantages like small device footprints and low production costs while providing the possibility to

manipulate fluids with better throughput and sensitivity than classical methods (Zhang *et al.*, 2017). In addition, modern microfabrication techniques like photolithography have been used to successfully manufacture microelectronics and micromechanical systems which has enabled the development of devices with hundreds of microfluidic operations. These developments have led to pneumatic microfluidic systems comprised of networks of tubes or channels through which air can freely flow. The free-flowing air provides the pressure to open or close a valve which represents a logical state paving the way for possibility for pneumatic microfluidic logic. Researchers like Jensen *et al.* (2007) have utilized this unique form factor of pneumatic logic to develop devices that function as complex digital logic circuits like full and multi-bit adders (Jensen *et al.*, 2007).

The developments in microfluidics and pneumatics mentioned above have been utilized in soft robot control; however, there is a considerable lack of solutions that are designed to accommodate multi-actuator systems with numbers of actuators in the 10s, 100s, or 1000s. Consider, for example, the approach adopted by Xu *et al.* (2020) involving the embedding of distributed fluidic circuitry into the soft structure of fluidic valves (Xu & Pérez-Arancibia, 2020). Their approach effectively utilizes fluidic logic by introducing a logic gate-based mapping, which can store discretized representations of the actuator's state to eliminate the need for electrical control. Although Xu *et al.*'s solution addresses a major problem with soft robot control, i.e., reliance on rigid electrical control, it fails to provide a scalable method to control multi-actuator soft robotic systems. This is because Xu *et al.*'s embedded switch-valves are not scalable for systems with multiple actuators and larger numbers of degrees of freedom. As the number of actuators and degrees of

freedom increases in a soft robotic system, the logic gate mapping gets exponentially more complex which makes fabricating the embedded fluidic circuitry for the soft switch-valves increasingly difficult. The “hard-coded” mappings also restrict the potential applications of the soft robot and require optimal pre-determined geometric parameters when designing the valves and the associated embedded circuitry, making the design and implementation process ineffective and restrictive.

Alternatively, Preston *et al.* (2019) present a more adaptable approach by utilizing custom soft pneumatic valves as bistable switches to develop basic logic gates like NOT, AND, and OR. The valve-based logic gates are then used to configure complex digital circuits like leading-edge detectors, digital-to-analog converters (DAC), toggle switches, etc. (Preston *et al.*, 2019). This demonstrates a significantly more flexible method that leverages combinational logic to develop control mechanisms for a wide variety of soft robotic systems and devices without having to depend on hard electronics. Although Preston *et al.*'s approach is more scalable in comparison with Xu *et al.*'s (2020) embedded switch-valve method, the proposed soft valves are bulky, difficult to fabricate, and provide limited response time. As identified by Preston *et al.*, the soft valves require a multi-step fabrication method that involves complex silicone molding and extensive thermal bonding. Moreover, each soft valve has a reported response time of 500 ms, which increases with the number of cascaded logic gates in the circuit, meaning more complex digital logic circuits have remarkably slower response times. Hence, to address problems with soft robot control there is a need for soft logic circuits that are comparatively easy to manufacture and perform more effectively.

These requirements can be fulfilled if Preston *et al.*'s large and complex soft valves are replaced with Grover *et al.*'s (2003) monolithic elastomer membrane valves to develop digital logic circuits for soft robot control. Grover *et al.* have successfully demonstrated that complex lab-on-chip microfluidic analyzers can be manufactured with relative ease by integrating the chips with their monolithic membrane valves. Grover's three-layer membrane valve design features etched glass wafers, a membrane, and (unlike Preston) a non-thermal bonding process which results in microfluidic devices that are small, relatively easy to produce, and of high performance (Grover *et al.*, 2003).

In addition to the application of monolithic membrane valves in analyzers, researchers like Jensen *et al.* (2007) have demonstrated their use in the development of digital logic circuits. Like Preston *et al.* (2019), Jensen *et al.* have developed digital logic gates AND, OR, NOT, NAND, and XOR using the monolithic membrane valves and in turn combined those logic gates to form 4-bit and 8-bit adders (Jensen *et al.*, 2007).

Although this approach is identical to that of Preston *et al.*, the added advantages of the monolithic membrane valves make Jensen's digital logic circuits easier to fabricate and have better performance. As reported by Jensen *et al.*, the logic gates developed using the monolithic membrane valves have response times of at most 250 ms which is twice as fast as Preston's soft valves.

Another notable application of the monolithic membrane valves which addresses all the previously mentioned problems with soft robot control is Huong *et al.*'s (2021) pneumatic random-access memory. Huong *et al.* have leveraged the normally-closed (NC) nature of the monolithic membrane valves and the associated latching functionality to create pneumatic memory which allows the control of soft robotic

systems with up to 8 independent actuators with the use of only 3 solenoid valves. The approach proposed by Huong *et al.* utilizes an identical fabrication approach to that of Grover *et al.* which makes the pneumatic random-access memory circuits easy to fabricate with remarkable performance. Additionally, the pneumatic memory circuits reduce the number of solenoid valves needed to drive soft robotic systems which in turn lessens dependence on rigid electrical components and provide a scalable way to control multi-actuator soft robots. Although Huong *et al.*'s work demonstrates potential for implementation in multi-actuator systems, it lacks explicit design and fabrication methodologies for soft robotic systems with 10s, 100s, or 1000s of independent actuators. Additionally, the work lacks characterization and evaluation of the pneumatic memory circuits which makes its scope in soft robot applications limited.

1.4. Objective

This research aims to fill the gap in the work of Grover, Jensen, and Huong *et al.* by designing, fabricating, and experimenting with pneumatic latched demultiplexer circuits implemented using the monolithic membrane valves that can provide control mechanisms for multi-actuator soft robotic systems. The pneumatic demultiplexer circuits proposed in this research are identical to the pneumatic random-access memory developed by Huong *et al.* (2021). Huong's pneumatic memory is a 3-bit pneumatic latched demultiplexer that can provide control for 8-actuator soft robotic systems. This research aims to expand that implementation to create a 4-bit pneumatic latched demultiplexer that can provide control for 16-actuator soft robots by using only 4 input solenoid valves. In addition, the goal of this research is to provide design, fabrication,

and testing frameworks to develop and implement n -bit pneumatic latched demultiplexers which can be used to control soft robotic systems with 2^n independent actuators.

Chapter 2

2. Concepts

To better describe the objectives and goals of this research, it is necessary to provide an overview of the concepts that are crucial and central to it. As an interdisciplinary project, this research utilizes ideas from both electronics and microfluidics. Since the main goal of this work is to develop pneumatic digital circuits, concepts from electronics are utilized to produce the required microfluidic devices. This section explores concepts from both disciplines to draw parallels and correlations between them.

2.1. Electronics-Microfluidics Analogy

To simplify the design, modeling, and simulation of complex microfluidic systems, researchers have developed electronic-fluidic analogies. These analogies draw similarities between concepts from electronics and microfluidics such as pressure and voltage, volumetric flow and electrical current, and fluidic and electrical resistance. This section explores these analogies and the correlation between them in order to establish a reference for the terminology and concepts used in this research.

The most fundamental analogy that is ubiquitously identified by researchers is the relationship between Ohm's law and Poiseuille's law. A simplified expression of Poiseuille's law can be easily correlated to Ohm's law, as both laws describe a linear relationship between two variables with a constant of proportionality, commonly

referred to as the resistance, which depends on the physical properties of the channel and the conductive medium, respectively (Perdigones *et al.*, 2014). Given below are the equations for both Ohm's law and Poiseuille's law to illustrate the correlation. Equation (1) shows Ohm's law, where R is the resistance, ρ is the electrical resistivity, A is the resistor cross-section, L_e is the resistor length, and I is the electrical current.

$$V = R \cdot I = \frac{\rho \cdot L_e}{A} \cdot I \quad (1)$$

Equation (2) shows Poiseuille's law, where ΔP is the pressure difference, R_m is the fluidic resistance, L_c is the channel length, μ is the viscosity, R is the channel radius, and Q is the volumetric flow rate.

$$\Delta P = R_m \cdot Q = \frac{8 \cdot L_c \cdot \mu}{\pi \cdot R^4} \cdot Q \quad (2)$$

Two other fundamental laws in the electrical domain are Kirchhoff's current law (KCL) which states that the sum of currents at a node is zero, and Kirchhoff's Voltage Law which states that the sum of voltages in a closed loop equals zero. Both KCL and KVL are given below in Equations (3) and (4) respectively:

$$\sum_{i=1}^n I_i = 0$$

(3)

$$\sum_{i=1}^n V_i = 0$$

(4)

The analogous fluidic law to KCL comes from the continuum mechanics of fluids, whereas that of KVL can be derived from Newtonian conservation of linear momentum of the fluid. The corresponding equations for both of the aforementioned laws are given below in Equations (5) and (6) respectively:

$$\sum_{i=1}^n Q_i = 0$$

(5)

$$\sum_{i=1}^n P_i = 0$$

(6)

2.2. Monolithic Membrane Valves

The microfluidic valves proposed for use in this project are Grover *et al.*'s normally-closed monolithic membrane valves. As mentioned before in the section 1.3. Prior Work, Monolithic membrane valves are comprised of three layers: a glass or polymethylmethacrylate (PMMA) layer containing etchings for control chambers and channels, another glass or PMMA layer containing etchings for input and output channels, and a PDMS membrane sandwiched between the two etched glass or PMMA layers. Fig. 2 shows the composition of the monolithic membrane valves in the form of a diagram. In a monolithic membrane valve, vacuum represents a logical 1 state (or HIGH or on) whereas atmospheric pressure represents a logical 0 state (or LOW or off).

When the control channel is kept open to atmospheric pressure the PDMS membrane stays in its resting position blocking the flow of air between the input and output channels. However, when vacuum is applied to the control channel, a pressure difference is created causing the PDMS membrane to be pulled into the control chamber, opening the valve, and allowing the flow between input and output channels (as shown in Fig. 2 below).

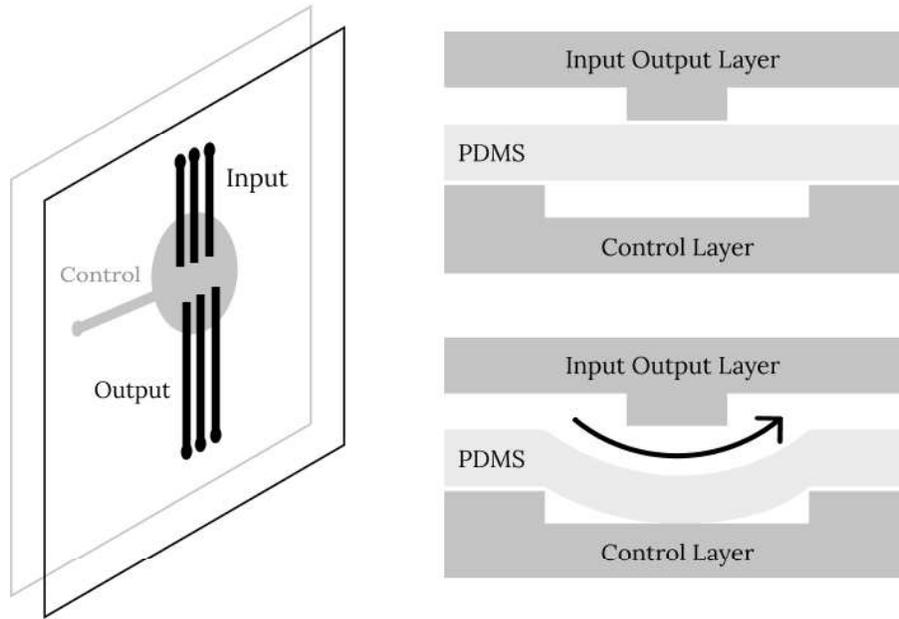


Fig. 2. Side view (left) and cross-sectional (right) view of the monolithic membrane valves, transistor-like elements for pneumatic digital circuits to illustrate their composition and working.

2.2.1. P-MOS Transistor

As transistors are the basic building blocks of electrical circuits, monolithic membrane valves are the fundamental building blocks of all the pneumatic digital circuits presented in this research. Moreover, the function of the normally-closed (NC) monolithic membrane valves can be directly compared to that of p-channel metal-oxide-semiconductor-field-effect transistors (p-MOSFET). Similar to how p-MOS transistors act as electrical voltage inverters, monolithic membrane valves can be configured in their elementary form to behave as pneumatic inverters (as shown in Fig. 3 below)

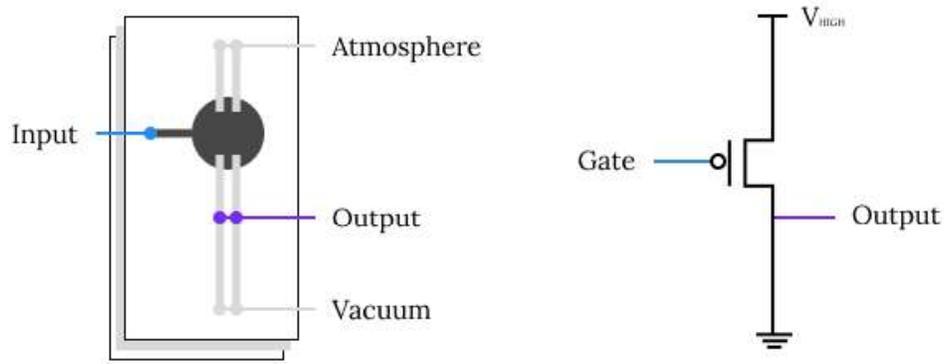


Fig. 3. Monolithic membrane valve pneumatic inverter (left) and a p-MOSFET electrical inverter (right).

The configuration of the monolithic membrane valve shown in Fig. 3 acts as a pneumatic inverter because when the control input is supplied vacuum (logical 1) the membrane is pulled into the control chamber opening the valve. This means that air flows from the top channel open to atmosphere to the bottom vacuum opening and the pressure at the output is equivalent to atmospheric pressure (logical 0). Whereas, when the input is kept open to atmosphere, the membrane stays in its closed position, which means the air flows from the bottom channel to the output channel, hence the output produces vacuum (logical 1).

2.2.2. D Latch

In addition to functioning as p-MOS transistors, monolithic membrane valves also behave like D latches. In electronics, a D latch is a digital circuit that can store one bit of information. In other words, a D latch is a basic memory element used to capture or memorize the logic state of a single input. The unique physical properties of the

monolithic membrane valves allow it to function like D latches. Due to the normally-closed nature of the monolithic membrane valves, they can remain sealed against pressure differentials even when disconnected from the pressure source. Hence, they can be used to trap input pressures as pneumatic logic state (Huong *et al.*, 2021). This unique latching functionality of the monolithic membrane valves proves to be incredibly useful in soft robot control, as the state of a certain actuator can be stored in the valves without the need for an active pressure source.

2.3. Demultiplexer

Since the goal of this project is to reduce the rigid electrical components in soft robot control mechanisms, one way to achieve this is by reducing the number of solenoid valves required to control the pneumatic actuators in soft robotic systems. To achieve this, the concept of demultiplexing is utilized in this work. In electronics, a demultiplexer is a device that takes an input and directs it to any one of a number of individual output lines based on its input select signals. A demultiplexer circuit with n select bits can select 2^n outputs. Hence, a pneumatic circuit that functions as a demultiplexer can control 2^n actuators using only n solenoid valves. This is an exponential improvement over conventional control systems which allow the control of only n actuators using n solenoid valves (see Fig. 1 in section 1.2. Problem). Fig. 4 shows a diagram of a pneumatic demultiplexer along with its data and select inputs and outputs.

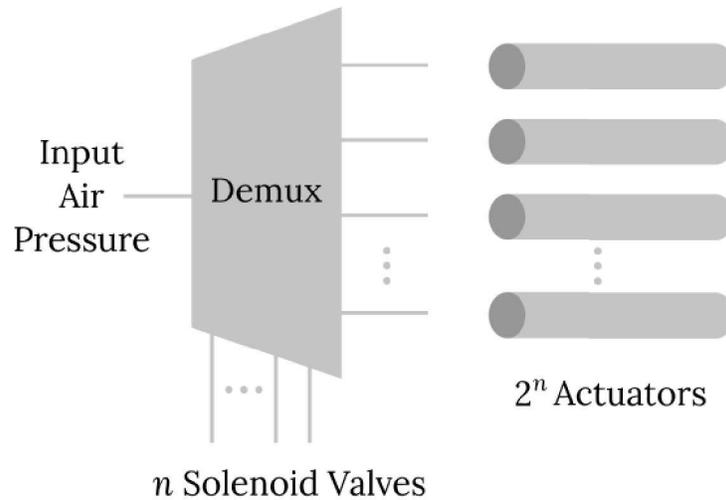


Fig. 4. Pneumatic demultiplexer with n input solenoid valves as select bits, input pressure as the data bit, and the 2^n actuators as outputs.

This research explores the design and development of the pneumatic demultiplexer shown in Fig. 4 by implementing and fabricating 4-bit and an arbitrary n -bit pneumatic demultiplexers which can provide control mechanisms for 16-actuator and 2^n -actuator soft robotic systems with minimal use of rigid electronics like solenoid valves, printed circuit boards (PCB), microcontrollers, etc.

Chapter 3

3. Methodologies

This chapter presents a design overview of the approach used to develop the pneumatic demultiplexer circuit using the monolithic membrane valves, the microfabrication methodologies used to manufacture the pneumatic chip, and the control processes used to drive the pneumatic circuit and test its functionalities.

3.1. Design Overview

Conventionally, electrical digital circuits can be designed using Boolean algebra and implemented using elementary logic gates such as AND, NAND, OR, NOR, etc. (Tokheim, 2017). The same process can be applied to the design of pneumatic digital circuits. In this work, pneumatic AND gates are used as the building blocks to develop the required pneumatic demultiplexer circuits. This section outlines the design of the pneumatic AND gate, developed using Grover *et al.*'s (2003) monolithic membrane valves, as well as the process that can be used to develop pneumatic demultiplexers using these pneumatic AND gates (which can be extrapolated to design other pneumatic digital circuits using other rudimentary pneumatic logic gates, as described in the work of Jensen *et al.* and Preston *et al.*).

The design of the pneumatic AND gate is simple, it contains two layers with a membrane in the middle (as explained in the section 2.2. Monolithic Membrane Valves).

Similar to how the monolithic membrane valves can be configured in their elementary form to function as pneumatic inverters (see Fig. 3 in section 2.2. Monolithic Membrane Valves), monolithic membrane valves can be configured to behave like pneumatic AND gates (as shown Fig. 5 below).

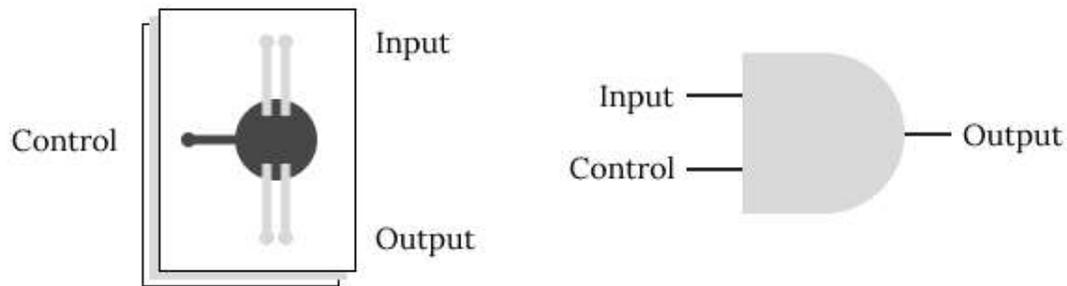


Fig. 5. Pneumatic AND gate with monolithic membrane valve (left). Equivalent electrical AND gate (with the same input and output labeling).

When vacuum (logical 1) is applied to the control channel of the pneumatic AND gate, suction is generated in the circular control chamber which pulls the middle-layer membrane into the chamber. This allows the flow of air from the input to the output. Hence when vacuum (logical 1) is also applied to the input channel, the output channel also produces vacuum (logical 1). Similarly, if the input channel is open to the atmosphere (logical 0), the output channel also produces a reading equivalent to the atmospheric pressure (logical 0). Alternatively, if the control channel is open to the atmosphere, the middle-layer membrane stays in its closed position restricting the flow of air from input to output which means that the output will always be at atmospheric

pressure regardless of the pressure being supplied to the input channel. Table 1 shows the theoretical truth table for the pneumatic AND gate summarizing its behavior.

Control Channel		Input Channel		Output Channel	
Pressure (Vacuum or Atmosphere)	Logical Values	Pressure (Vacuum or Atmosphere)	Logical Values	Pressure (Vacuum or Atmosphere)	Logical Values
A	0	A	0	A	0
A	0	V	1	A	0
V	1	A	0	A	0
V	1	V	1	V	1

Table 1. Theoretical truth table of the pneumatic AND gate representing the behavior of the pneumatic circuit with pressure values and the equivalent logical values, where “A” represents atmospheric pressure and “V” represents vacuum pressure.

The pneumatic AND gate described above can be used to create pneumatic digital circuits by connecting the pneumatic AND gates together, similar to how electrical digital circuits can be created by combining electrical logic gates. Consider, for example, a 1-bit electrical demultiplexer which can be implemented using two AND gates and a negated input. Its pneumatic counterpart can be implemented in the same way by using the pneumatic AND gate. Fig. 6 shows both the electrical and pneumatic 1-bit demultiplexer developed using the AND gates.

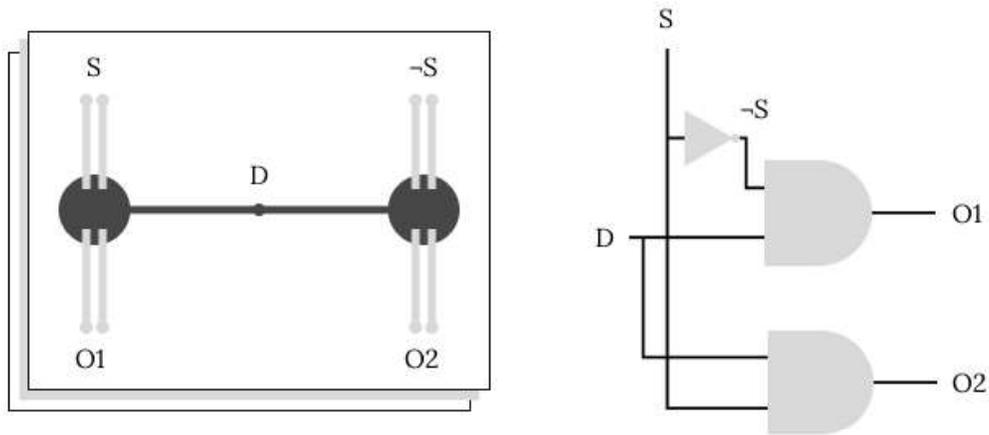


Fig. 6. 1-bit pneumatic demultiplexer circuit (left) and equivalent electrical demultiplexer circuit (right) developed using two AND gates and a negated input.

The same process can be applied to develop pneumatic digital circuits like n -bit demultiplexers, AND gates, full and half adders, etc. This is the process used by Huong *et al.* (2021) to develop the 3-bit demultiplexer discussed extensively in this paper (shown in Fig. 7 below along with its electrical counterpart).

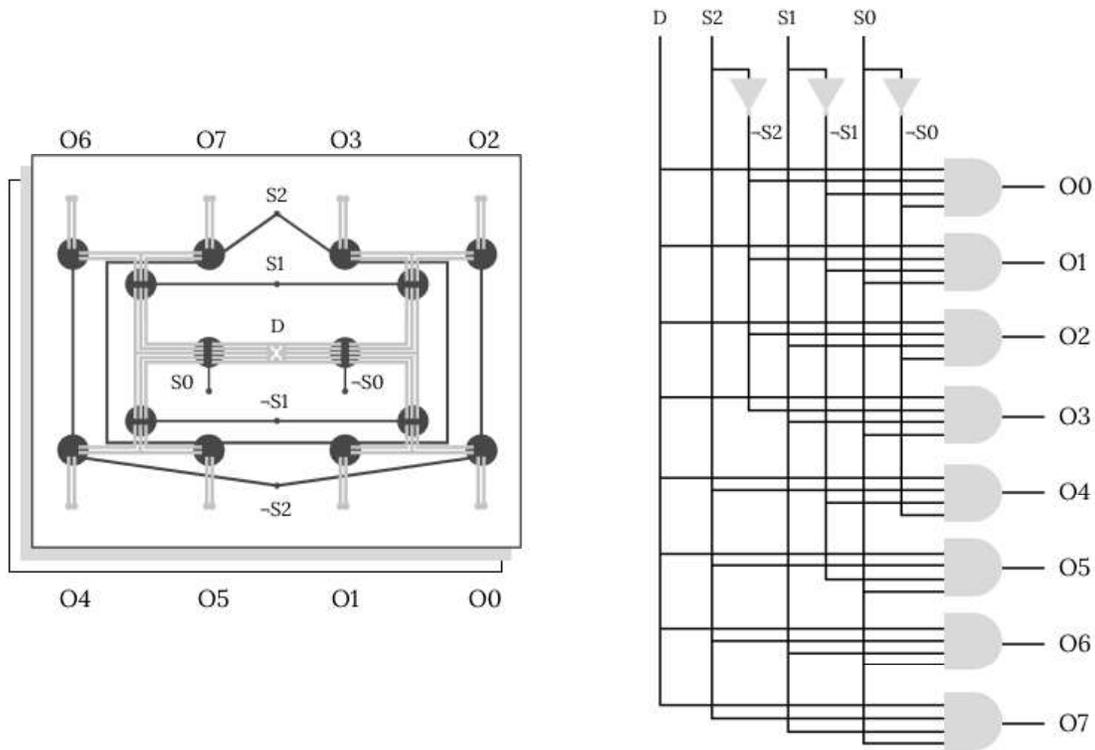


Fig. 7. Huong et al.'s 3-bit pneumatic demultiplexer circuit (left) and equivalent electrical demultiplexer circuit (right).

3.2. Controlling the Pneumatic Demultiplexer

The control mechanism of the pneumatic demultiplexer is similar to that of its electrical counterpart. Like an electrical demultiplexer, an n -bit pneumatic demultiplexer has one data channel, n -select channels, and 2^n -output channels. The data channel is supplied with the pressure that is needed at the output channels. For instance, if -50 kPa is the required output, the data channel should be supplied -50 kPa. Similarly, the select channels are supplied either vacuum, which is equivalent to a logical 1, or is left open to

the atmosphere which is equivalent to a logical 0 depending on the output channel to be actuated. The negated select channels should always be supplied the opposite of their positive select counterparts. For example, if select channel S1 is supplied with atmospheric pressure, negated select channel $\neg S1$ should receive a vacuum input.

Consider the 3-bit pneumatic demultiplexer shown in Fig. 7 (section 3.1. Design Overview) that contains 3 select channels, 8 output channels (2^3), and a data channel. Assume a vacuum response of -50 kPa is needed at the output O2 (3rd output) to actuate a vacuum-operated soft robotic actuator. To accomplish this the following steps should be taken. First, the data channel should be supplied -50 kPa to get an output response equivalent to -50 kPa. Secondly, a select configuration of 010_2 is needed to actuate output O2 since 010_2 in binary (base 2) is equivalent to 2_{10} in decimal (base 10). In a 3-bit electrical demultiplexer, a select configuration of 010_2 would result in the 3rd output being “selected”, hence the same configuration can be used in a pneumatic demultiplexer to select the 3rd output or the O2 channel. In order to set a select configuration of 010_2 select bit S0 should be left open to atmosphere or digital 0, $\neg S0$ should be supplied vacuum or digital 1, S1 should receive vacuum (digital 1), $\neg S1$ open to atmosphere (digital 0), S2 open to atmosphere (digital 0), and $\neg S2$ should receive vacuum (digital 1). Fig. 8 shows the pressure inputs needed to actuate output O2 in a pneumatic demultiplexer along with the digital inputs needed to get the equivalent response in an analogous electrical demultiplexer.

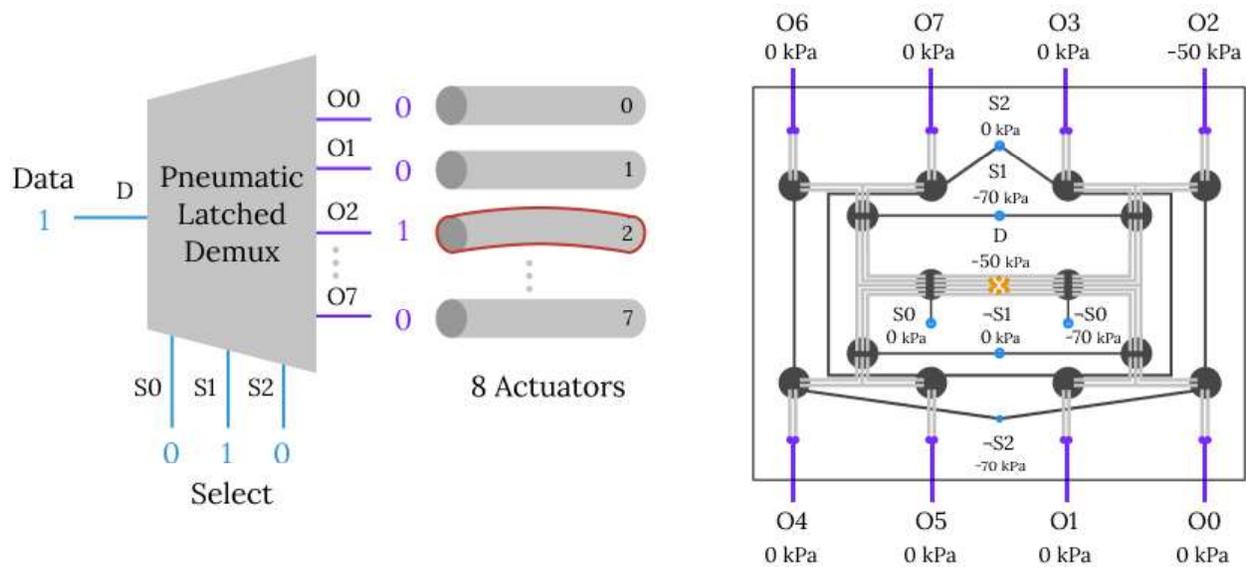


Fig. 8. Pressure inputs needed to select the output O2 or 3rd output with a pressure response of -50 kPa using a 3-bit pneumatic demultiplexer (right). Digital select configuration to select output O2 or 3rd bit using a 3-bit electrical demultiplexer (left).

3.2.1. Driver Hardware Setup

Since the proposed n -bit pneumatic demultiplexer contains $2n+1$ individual channels (1 data, n positive select, and n negated select channels), manually switching the inputs is incredibly inefficient. Hence in order to drive inputs of the pneumatic demultiplexer, 2-position 4-way solenoid valves were used. Each select bit (which includes both the positive and negated select channels) was connected to its own individual solenoid valve, i.e., an n -bit pneumatic demultiplexer would require n solenoid valves to drive 2^n outputs. As mentioned in section 2.3. Demultiplexer, this is an exponential improvement over the 2^n solenoid valves that would be required if the solenoid valves

were directly connected to each of the 2^n actuators. Each solenoid valve contains two input ports, one of which is connected to a vacuum source and the other one is left open to the atmosphere. The two output ports of each solenoid valve are connected to the positive and negated select channel of a select bit. When an electrical current is applied to the solenoid valves, the outputs are reversed. For example, if in its inactive state (with no electrical current) the positive select channel is being supplied vacuum and the negated select channel is open to atmosphere, in its active state (with electrical current) the positive select is open to atmosphere and the negated channel is supplied vacuum. Hence by applying electrical current to the respective solenoid valve, a select bit can be set to vacuum (digital 1) or to atmosphere (digital 0), allowing full control of the pneumatic demultiplexer.

The vacuum for the solenoid valves was provided using a right-angle flow manifold. A DC power supply was used for the 24V DC input needed to drive the solenoid valves. Relays were used along with an Arduino Leonardo microcontroller to provide the input signals to the solenoid valves via a software interface (see *Generalized Controller Software*). Lastly, to record pressure response at the output channels, an I2C pressure sensor was used along with a digital pressure gauge. Fig. 9 shows the hardware setup needed to drive the pneumatic demultiplexer circuits and Table 2 shows the components needed to assemble the hardware setup.

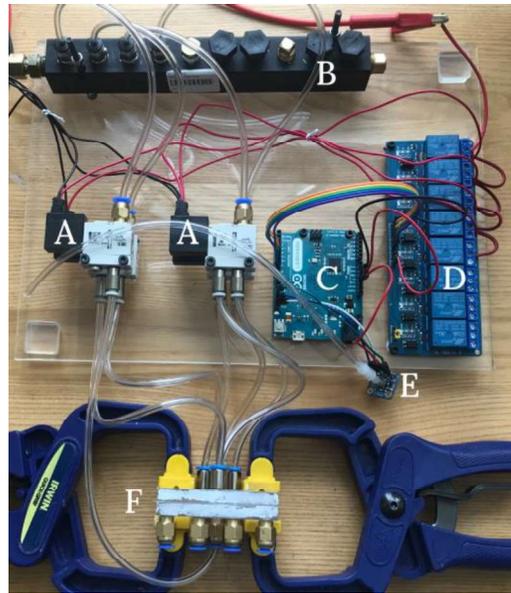
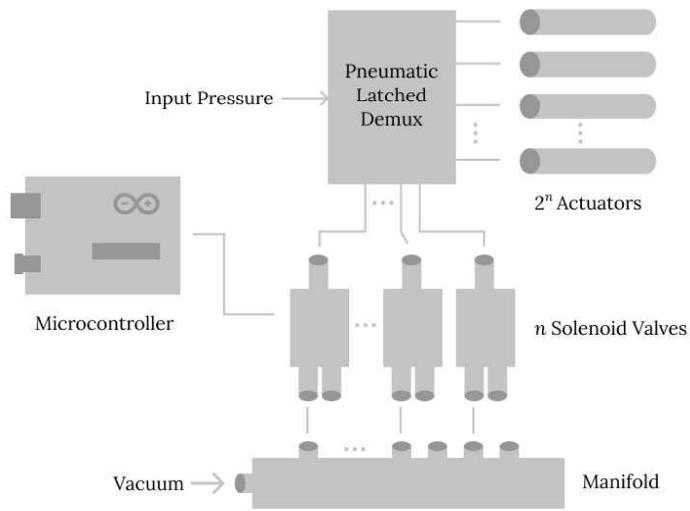


Fig. 9. Hardware setup needed to drive n -bit pneumatic demultiplexers (left). Hardware setup assembled to drive 3-bit and 4-bit pneumatic demultiplexers.

ID	Component	Qt.	Description
A	SMC VQD1121-5L-M5	4	2 position 4 way solenoid valve
	McMaster-Carr 7880T389	40	4 mm OD M5 push-to-connect fitting
B	McMaster-Carr 5469K191	1	Aluminum right-angle flow manifold
	Tailonz PC4-N1	5	4mm OD 1/8" NPT push-to-connect fitting
C	Arduino Leonardo	1	Microcontroller
D	SunFounder TS0012	1	5V 8 channel relay shield
E	Adafruit MRPLS	2	Ported pressure sensor breakout
N/A	McMaster-Carr 5233K112	50 ft.	2mm ID 4mm OD soft PVC plastic tubing

Table 2. Components needed to assemble hardware setup to drive 3 and 4-bit pneumatic demultiplexers.

3.2.2. Generalized Controller Software

In order to make testing and controlling the pneumatic demultiplexers intuitive and efficient, a controller software was developed. The controller software's comprehensive interface allows users to control solenoid valves connected to the select channels of the pneumatic demultiplexer, plots and shows pressure response as recorded via the I2C pressure sensor and allows users to capture pressure data in the form of a comma-separated-value (.csv) file. Fig. 10 shows a screenshot of the controller interface.

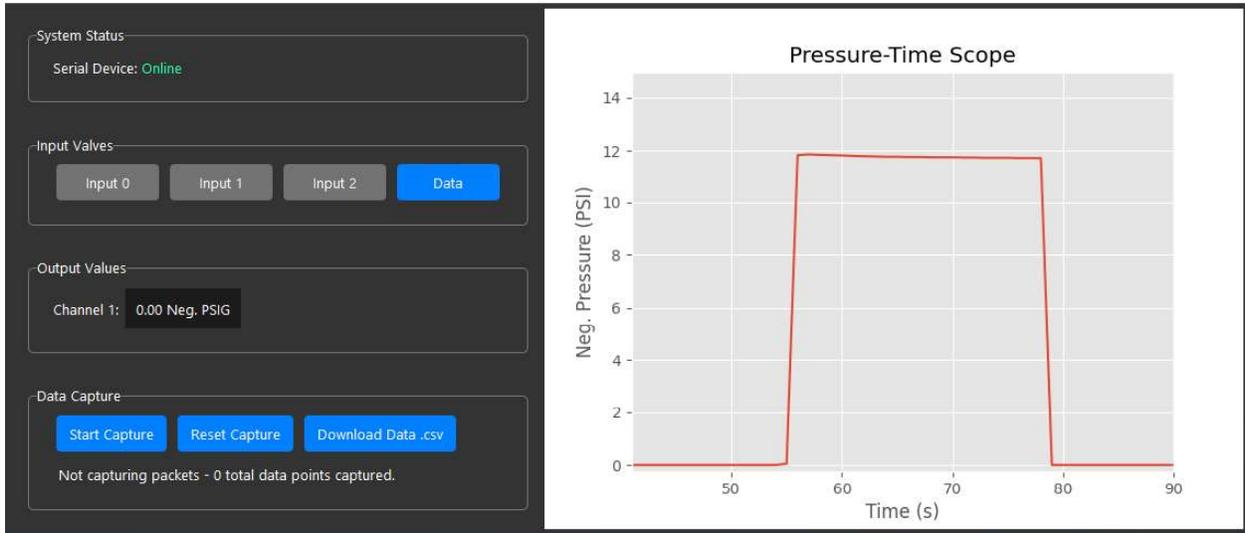


Fig. 10. Screenshot of the controller interface for driving and testing the pneumatic demultiplexers.

As shown in Fig. 10, the left panel of the controller’s user interface includes the system status to show whether the microcontroller is online or offline, toggle buttons to control solenoid valves connected to select and data channels, pressure output values recorded in real-time via the pressure sensor connected to the microcontroller, and buttons to capture and download pressure data in the form of a comma-separated-value (.csv) file.

The software was developed in a general manner i.e., the program on the microcontroller is independent of the controller software which runs on the host computer. This was done to ensure that the software is versatile enough to accommodate a wide range of driver hardware and experimental setups. Since the controller software uses serial communication to communicate with the

microcontroller, setups with other types of microcontrollers can be easily integrated with the controller software.

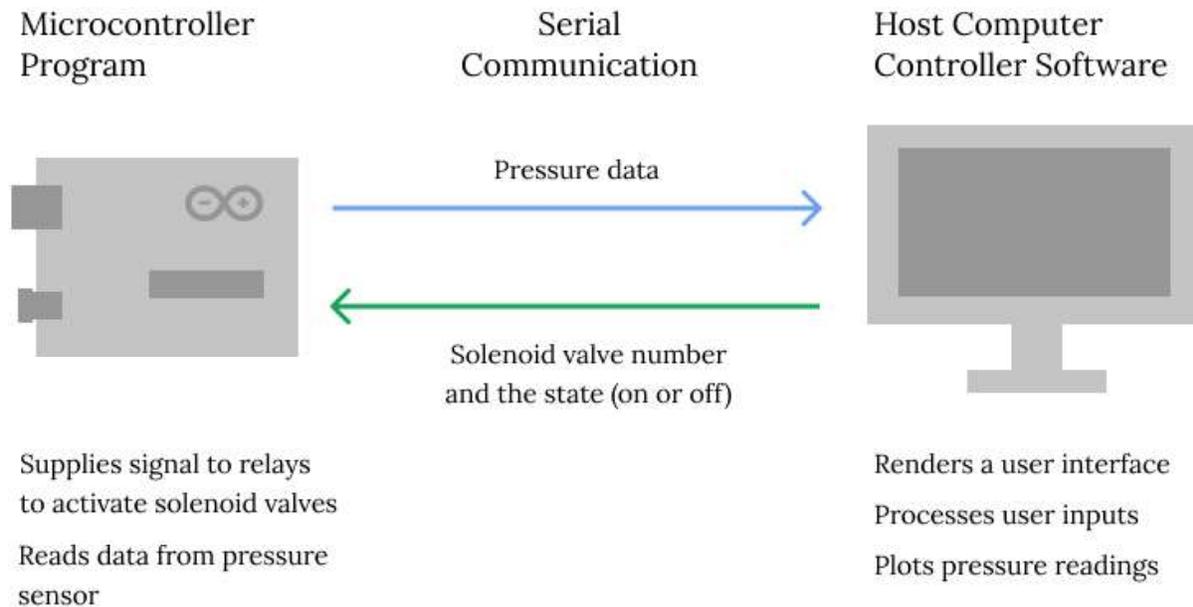


Fig. 11. Overview of the generalized nature of the controller software.

As shown in Fig. 11, on the microcontroller side the program supplies a signal to the relays when a message with the respective instruction is received via serial communication. Additionally, the microcontroller program continually reads and writes pressure data from the sensor to the serial port. On the host computer side, the controller software displays the user interface and records user interactions. Based on the interactions, the program writes instructions to the serial port for the microcontroller. Moreover, the host computer reads the pressure data from the serial port and plots it in real-time on the user interface. Table 3 shows the communication protocol used by the microcontroller program and the controller software.

Message	Description
BEGIN	Calibrate pressure sensor and begin sending sensor data
ACTIVE	Pressure sensor has been successfully setup
ERROR	Pressure sensor not responding or failed to setup
{:id}	Turn on the solenoid valve with id
{:id + 5}	Turn off the solenoid valve with id
END	Stop sending sensor data

Table 3. Protocol for the generalized controller software to establish communication between the microcontroller and host computer.

Hence by following the protocol shown in Table 3, any microcontroller with serial communication capability can utilize the generalized controller software to drive and test pneumatic demultiplexer circuits or control soft robotic systems driven by pneumatic demultiplexers.

3.3. Microfabrication

In order to fabricate the pneumatic demultiplexers, two different methods were investigated. The first method replicates the technique that is common in the field of microfluidics and lab-on-chip manufacturing to fabricate normally-closed (NC)

microfluidic valves (Scott & Ali, 2021). The second method is unique to this work and was developed specifically to better suit the goals and scope of this thesis, i.e., making microfabrication more accessible and feasible. This section explores each method in detail, including comprehensive accounts of the manufacturing process along with experimental results obtained from functionality tests.

3.3.1. Micromilling and Direct Bonding Method

For this method, the design of the pneumatic chip (shown in Fig. 7 in the section 3.1. Design Overview) was converted into a computer-aided design (CAD) drawing using Dassault Systèmes SOLIDWORKS shown in Fig. 12 below.

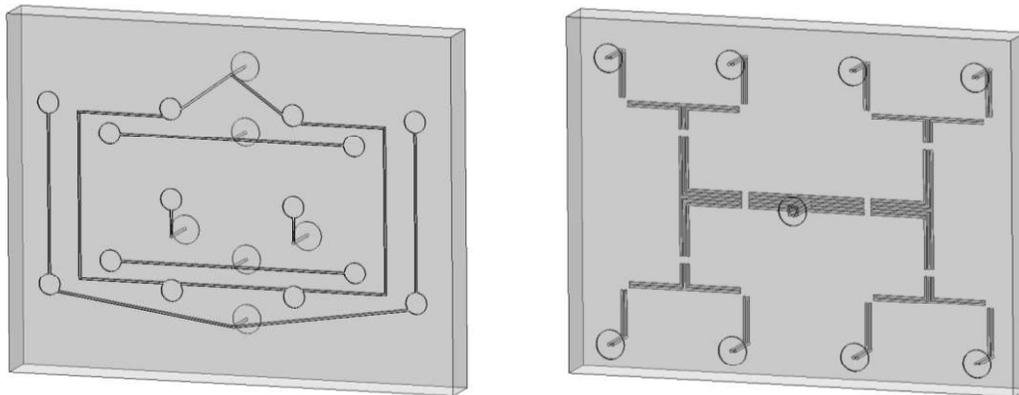


Fig. 12. CAD drawings of the 3-bit pneumatic demultiplexer circuit developed for CNC milling the circuit.

The drawings were then used to machine the microvalves and microchannels onto two 9 mm thick polymethylmethacrylate (PMMA) wafers (McMaster-Carr 8560K354, acrylic sheet) by utilizing micromilling techniques using a CNC machine. Additionally, M5 threads were tapped onto the unetched side of the PMMA for the through holes. The

PMMA wafers were then submerged in 99% Isopropyl Alcohol (IPA) for 20 minutes to clean any residues left from the milling process. After drying each PMMA part, the etched surface of each wafer was plasma treated for 1 minute. This was done because plasma treatment creates polar functional groups on the surface of PMMA that enhance adhesion to PDMS (Tan *et al.*, 2010). The plasma-treated PMMA parts were then bonded together using a 0.25 mm polydimethylsiloxane (PDMS) membrane (BISCO HT-6240, PDMS sheet) such that the parts create a PMMA-PDMS-PMMA sandwich. The parts were then clamped together for 1 hour to allow the PMMA-PDMS bond to strengthen. Lastly, push-to-connect fittings (McMaster-Carr 7880T389, 4 mm OD M5 fitting) were screwed into the threads and the pneumatic demultiplexer chip was finalized. Fig. 13 schematic summary of the manufacturing process and Fig. 14 shows a photograph of the pneumatic demultiplexer chip produced using the micromilling and bonding method.

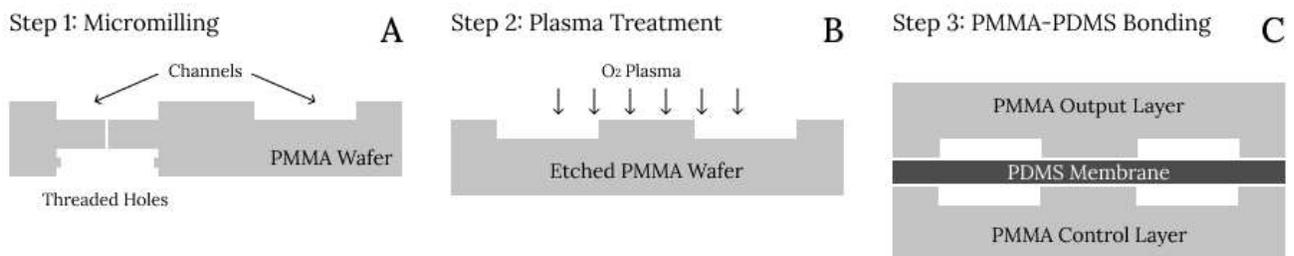


Fig. 13. Micromilling and direct bonding fabrication technique. (A) The first step involves micromilling channels and valves and tapping threaded holes onto PMMA. (B) The second step involves plasma treatment. (C) The third step involves sandwiching the PMMA-PDMS-PMMA layers.

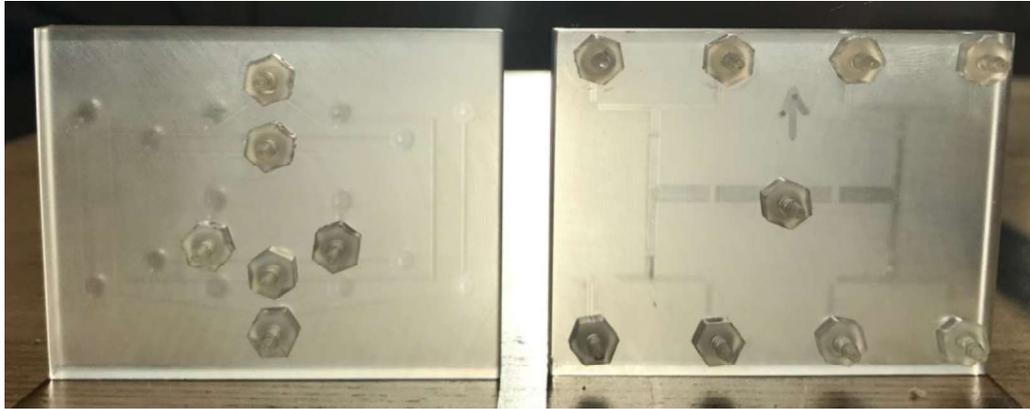


Fig. 14. 3-bit pneumatic demultiplexer chip fabricated using the micromilling and bonding technique.

The resulting pneumatic demultiplexer chip's functionality was tested using the approach outlined in the section 3.2. Controlling the Pneumatic Demultiplexer, i.e., each output was tested by supplying vacuum (digital 1) or atmosphere (digital 0) to its respective select channels. The data channel was supplied with a 72.8 kPa vacuum (digital 1). Given below in Table 4 is the truth table showing the test results for the 3-bit pneumatic demultiplexer fabricated using the micromilling and direct bonding technique.

Select Channels (Neg. kPa)			Output Channels (Neg. kPa)							
S0	S1	S2	O0	O1	O2	O3	O4	O5	O6	O7
75.7	75.5	73.2	0.1	0.4	0.1	0.0	0.2	0.2	0.4	14.1
75.7	75.5	0.0	0.0	0.1	0.3	0.1	0.2	0.1	16.3	0.5
75.7	0.0	73.2	0.2	0.0	0.5	0.7	0.3	1.7	0.3	0.0
75.7	0.0	0.0	0.5	0.8	0.6	0.8	0.0	0.2	0.1	0.3
0.0	75.5	73.2	0.2	0.0	0.0	24.9	0.1	0.0	0.2	0.1
0.0	75.5	0.0	0.3	0.3	19.4	0.0	0.2	0.3	0.1	0.0
0.0	0.0	73.2	0.4	14.9	0.3	0.1	0.3	0.1	0.0	0.0
0.0	0.0	0.0	15.6	0.5	0.0	0.0	0.1	0.2	0.1	0.5

Table 4. Experimental truth table for the functionality test of the 3-bit pneumatic demultiplexer fabricated using the micromilling and direct bonding technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned.

Although this technique is widely used in the fabrication of microfluidic lab-on-chip devices with NC valves, in the case of this project it was proved to be only partially successful. As is evident from Table 4, the pneumatic demultiplexer produced using the micromilling and direct bonding technique functions reasonably well except for 2 outputs (O4 and O5 channels). However, the failure of 2 outputs and a significant

pressure drop between the data channel and output channels of 81.6% is not acceptable since one of the goals of this thesis is to support real-world applications of soft robotics. A consistent and sufficient pressure response is needed on all outputs to drive the pneumatic actuators used in soft robots.

The reason for the observed failures can be attributed to leaks in the pneumatic chip which were recorded in output channels. For instance, consider the first select channel configuration in the truth table (Table 4) where S₀, S₁, and S₂ were supplied vacuum (111₂) which should result in output O₇ (111₂ = 7) being actuated (outputting vacuum or digital 1) and the rest of the outputs should produce atmosphere (digital 0). However, most outputs produce small amounts of vacuum due to leaks. This pattern can be seen for all select configurations. Moreover, further investigation revealed larger pressure drops over time due to the reversible nature of the PMMA-PDMS bond. Hence, it was concluded that the reason for the leaks is due to the weak PMMA-PDMS bonding.

Despite being a tested and used method, this technique requires specialized equipment like CNC machine and plasma treater and technical skills like micromilling which makes this method significantly less feasible. Moreover, this process is prone to error because the material properties of the PMMA can significantly impact the PMMA-PDMS bonding. These properties vary between various types of readily available PMMA and can be unintentionally altered during the micromilling process, making this technique less consistent.

3.3.2. Laser Engraving and Adhesive Method

The fabrication method discussed in the previous subsection 3.3.1. Micromilling and Direct Bonding Method was significantly modified by researchers at the University of Michigan in order to develop a technique that can be used to feasibly produce pneumatic demultiplexer chips with strong PMMA-PDMS bonding (Gillespie, 2023). For this method, the design of the pneumatic chip was converted into a 2-dimensional vector graphic using Adobe Illustrator (shown in Fig. 15 below).

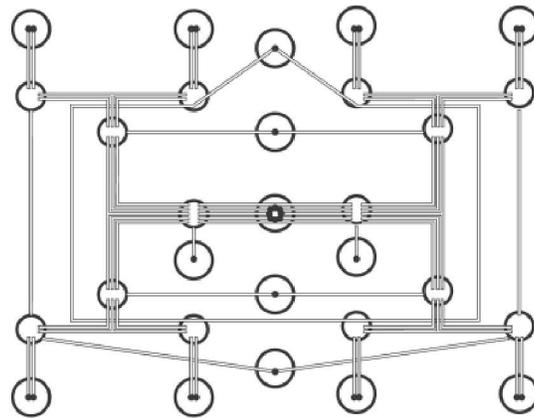


Fig. 15. 2-dimensional vector graphic of the 3-bit pneumatic demultiplexer circuit developed for laser cutting.

The original design of the 3-bit pneumatic demultiplexer by Huong *et al.* (2021) was altered for this method. Distances between the channels were increased by 0.2 mm and the valve diameter was increased to 4mm, to ensure that the manufacturing done with this method would not cause any deterioration of the features of the 3-bit pneumatic demultiplexer. Double-sided adhesive tape (3M 468MP, clear transfer tape) was applied to the surface of two 9 mm thick PMMA pieces (McMaster-Carr 8560K354, acrylic

sheet). The vector graphic was then used to laser engrave the microchannels and microvalves to the surface of PMMA wafers with the adhesive using a 75W industrial laser cutter. A combination of rastering (engraving) and vectoring (cutting) laser settings were utilized to get channels and valves with depths of approximately 0.8 mm and widths of approximately 0.3 mm. After laser engraving, M5 threads were tapped onto the unetched side of each PMMA wafer as channels and channels of the pneumatic demultiplexer chip. Once the manufacturing was completed, the PMMA parts were cleaned using pressurized air to clear any obstructions in the channels or valves caused during the laser cutting process. After cleaning, the tapes on the surface of both PMMA wafers were peeled exposing the adhesive and the 0.25 mm PDMS membrane (BISCO HT-6240, PDMS sheet) was sandwiched between the two etched PMMA pieces. Lastly, push-to-connect fittings (McMaster-Carr 7880T389, 4 mm OD M5 fitting) were screwed into the threads and the pneumatic demultiplexer chip was finalized. Fig. 16 shows a schematic summary of this fabrication process and Fig. 17 shows a photograph of the pneumatic demultiplexer chip produced using the laser engraving and adhesive method.

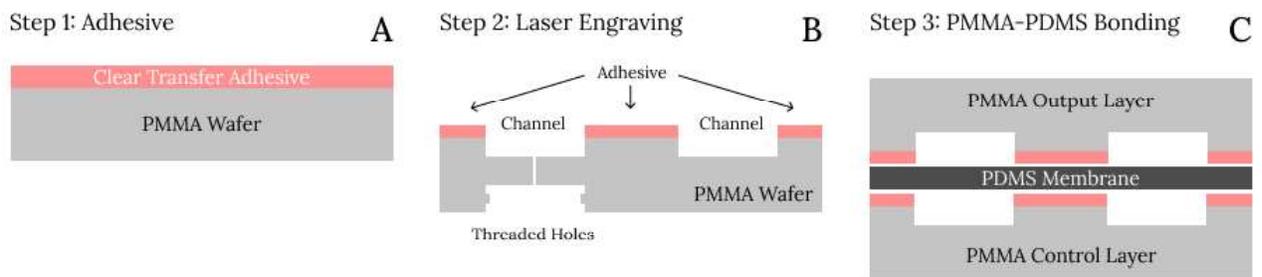


Fig. 16. Laser engraving and adhesive fabrication technique. (A) The first step involves applying double-sided adhesive tape to each PMMA piece.

(B) The second step involves laser engraving channels and valves and

tapping threaded holes onto PMMA. (C) The third step involves removing the tape and sandwiching the PDMS to the PMMA with the adhesive.

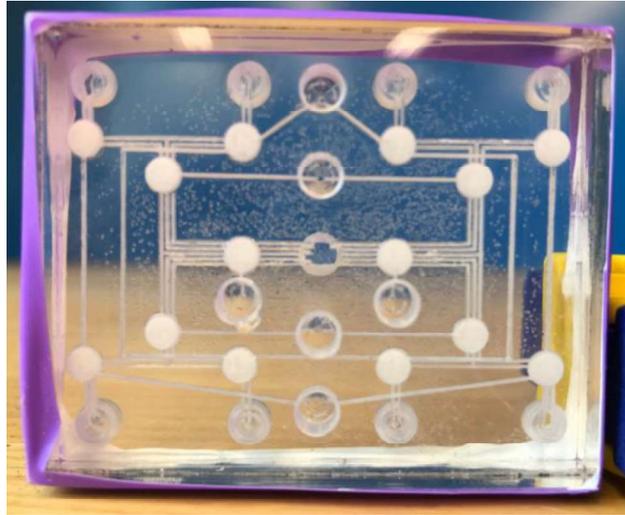


Fig. 17. 3-bit pneumatic demultiplexer chip fabricated using the laser engraving and adhesive method.

The resulting pneumatic demultiplexer chip's functionality was tested using the approach outlined in the section 3.2. Controlling the Pneumatic Demultiplexer i.e., each output was tested by supplying vacuum (digital 1) or atmosphere (digital 0) to its respective select channels. The data channel was supplied with a 77.3 kPa vacuum (digital 1). Given below in Table 5 is the truth table showing the test results for the 3-bit pneumatic demultiplexer fabricated using the laser engraving and adhesive technique.

Select Channels (Neg. kPa)			Output Channels (Neg. kPa)							
S0	S1	S2	O0	O1	O2	O3	O4	O5	O6	O7
77.0	75.8	76.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	64.9
77.0	75.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	63.4	0.0
77.0	0.0	76.8	0.0	0.0	0.0	0.0	0.0	66.9	0.0	0.0
77.0	0.0	0.0	0.0	0.0	0.0	0.0	71.2	0.0	0.0	0.0
0.0	75.8	76.8	0.0	0.0	0.0	73.7	0.0	0.0	0.0	0.0
0.0	75.8	0.0	0.0	0.0	73.2	0.0	0.0	0.0	0.0	0.0
0.0	0.0	76.8	0.0	75.7	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	74.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 5. Experimental truth table for the functionality test of the 3-bit pneumatic demultiplexer fabricated using the laser engraving and adhesive technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned.

The laser engraving and adhesive method is a considerable improvement over the micromilling and direct bonding method. As evident in Table 5, the improved pneumatic chip has virtually no leaks as inactive *output* channels have readings in the range of 0 - 0.01 kPa. Moreover, all outputs work as expected and the pressure drop between the data channel and output channels is reduced to 8.9%, a significant

improvement over the 81.6% pressure drop recorded in the pneumatic demultiplexer fabricated using the micromilling and direct bonding technique.

This method improves the previous technique by introducing an adhesive to improve the PMMA-PDMS bonding. Hence, plasma treatment, extensive cleaning, and smoothing of the PMMA surface are no longer required to obtain an irreversible PMMA-PDMS bond. However, using adhesive tape hinders the use of a CNC mill which is why laser engraving was used. Although laser cutting is considerably less precise when compared to micromilling, laser cutters are more readily available and are significantly easier to use, making this technique more feasible. Moreover, despite being less precise, laser cutters can easily produce the widths and depths required for the pneumatic chips included in this work.

3.4. Improvements

This section describes the methodologies utilized to improve the pneumatic demultiplexer developed by Huong *et al.* (2021). In addition to the refinement of the fabrication process (explored in the previous section 3.3. Microfabrication), techniques were developed to improve the volumetric flow rates and to increase the number of outputs (to create the next iteration 4-bit and the arbitrary n -bit pneumatic demultiplexer). Each of these processes is explored in detail in this section with comprehensive descriptions along with experimental results to quantify the improvements that were made.

3.4.1. Channel Scaling

Huong *et al.*'s pneumatic demultiplexer circuits produce volumetric flow rates of approximately 0.8 L/min. Since the objective of this research is to develop a system that can support a wide variety of soft robotics applications, the flow rates needed to be improved. A simple method to increase volumetric flow rates is to physically enlarge the dimensions of the microfluidic channels (Gallardo Hevia *et al.*, 2022). The Hagen-Poiseuille equation shown in Equation (7), where ΔP is pressure drop, L is channel length, Q is flow rate, μ is viscosity, and R is channel radius, defines the relationship between the channel radius and the volumetric flow rate.

$$\Delta P = \frac{8 \cdot L \cdot Q \cdot \mu}{\pi \cdot R^4} \quad (7)$$

It is important to note that the Hagen-Poiseuille equation gives the pressure drop in an incompressible and Newtonian fluid in laminar flow flowing through a cylindrical pipe. Although the conditions in the aforementioned definition do not apply to the pneumatic demultiplexer, the model can be used to approximate fluidic relationships for the pneumatic devices in this work. Hence, the Hagen-Poiseuille equation can be used to support the hypothesis that an increase in channel width or depth should correspond to an increase in volumetric flow rate. Since widening the channel would require the redesign of the pneumatic chip, the depth of the channels was increased to study the effects on flow rates. Instead of fabricating the full 3-bit pneumatic demultiplexers with varying channel depths, individual pneumatic AND gates were manufactured, in order

to keep the process simple and to conserve time. Four different pneumatic AND gates (from the pneumatic AND gate design shown in Fig. 5 in the section 3.1. Design Overview) were produced by utilizing the laser engraving and adhesive fabrication technique. The raster/engraving laser power was varied in order to obtain pneumatic chips with varying channel depths (see the subsection 3.3.2. Laser Engraving and Adhesive Method). Photographs of the pneumatic AND gates produced using this process are shown in Fig. 18 below.



Fig. 18. Pneumatic AND gates fabricated using laser engraving and adhesive method with varying channel depths of 2.7 mm, 1.5 mm, 0.8 mm, and 0.5 mm (from left to right).

The resulting pneumatic AND gates were tested individually using a flow meter. Given below is the table (Table 6) that shows the laser power settings used for each pneumatic AND gate along with channel depths produced and the recorded flow rates.

Laser Power (%)	Channel Depth (mm)	Volumetric Flow Rate (L/min)
75	0.5	1.0
50	0.8	1.8
25	1.5	2.5
15	2.7	4.3

Table 6. Flow rates recorded for pneumatic AND gates manufactured using different laser power settings with increasing channel depths.

The relationship between the channel depth and flow rate is better illustrated graphically. Fig. 19 shows the flow rate plotted against channel depth showing a linear increase in flow rate with an increase in channel depth.

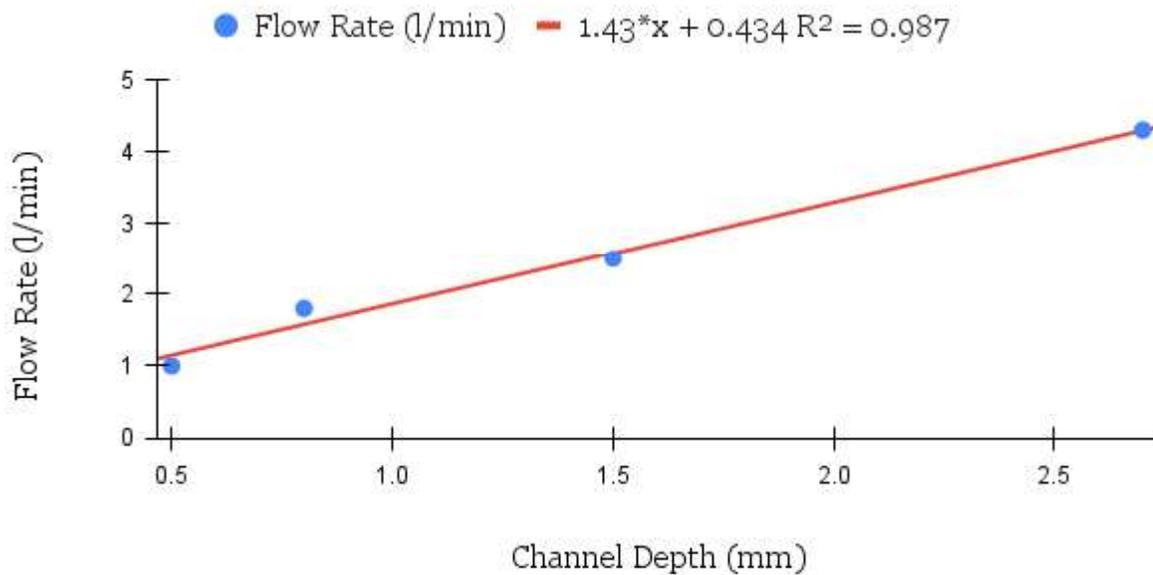


Fig. 19. Plot of volumetric flow rate against channel depth for pneumatic AND gates with varying channel depths illustrating the effect of channel depth on volumetric flow rate.

3.4.2. Branching

An objective of this research is to provide control mechanisms for multi-actuator soft robots. Hence it is important to devise a scalable method to develop n -bit pneumatic demultiplexers. The section 3.1. Design Overview describes one possible method for developing n -bit pneumatic demultiplexers; however, the process outlined in that section requires the use of Boolean algebra and pneumatic and electrical circuit design techniques to successfully develop the required pneumatic chips. Alternatively, a more scalable and feasible technique is branching, which involves the modification of the $n-1$ -bit pneumatic demultiplexer design to create the n -bit pneumatic chip. The process of

branching involves replacing the output channels in the $n-1$ -bit pneumatic demultiplexer design with a pair of monolithic membrane valves such that the output channel “branches” into two different output channels. All the newly added valves are then connected together such that the left valves are controlled using the new input channel and the right valves are controlled using the new negated input channel. Fig. 20 shows the 4-bit pneumatic demultiplexer developed using this method as an example of the branching technique in practice.

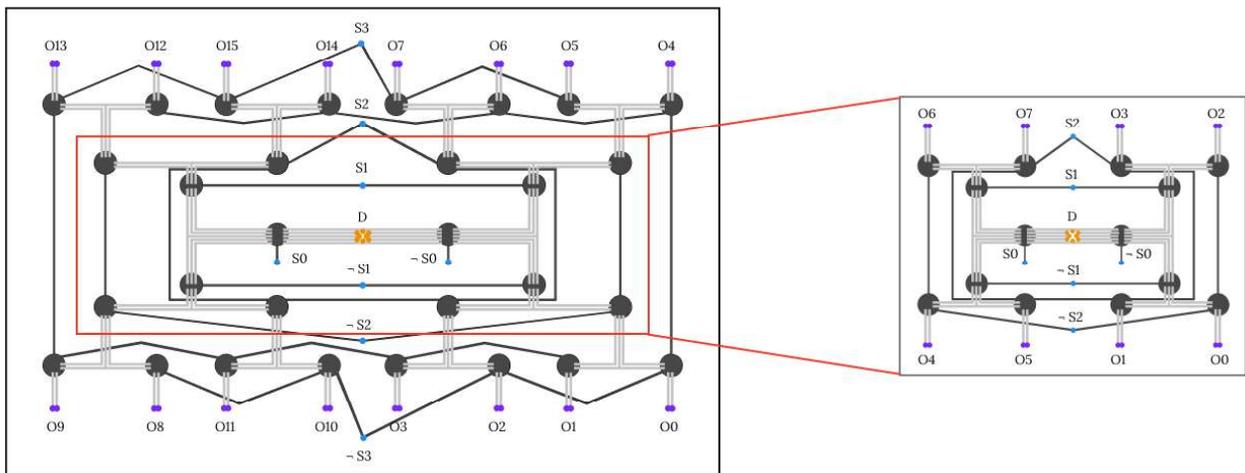


Fig. 20. 4-bit pneumatic demultiplexer designed using the branching technique. The red-bordered area shows the 3-bit pneumatic demultiplexer design used to develop the 4-bit version.

As evident in Fig. 20, the 4-bit (n -bit) pneumatic demultiplexer design is simply an extension of the 3-bit ($n-1$ -bit) design. As mentioned before, the output channels of the 3-bit pneumatic demultiplexer are now branched using a pair of parallel channels and two monolithic membrane valves. The monolithic membrane valves are connected in

such a way that every single left valve is controlled via the added select channel S_3 , similarly all the right valves are controlled using the new negated select channel $\neg S_3$. Hence, a single new select channel allows the control of 8 additional outputs. The same branching procedure can be applied to the 4-bit pneumatic demultiplexer to design a 5-bit chip and the process can be replicated for any arbitrary n -bit demultiplexer design.

The 4-bit pneumatic demultiplexer design developed using the branching technique was fabricated using the laser engraving and adhesive technique and is shown in Fig. 21 below.

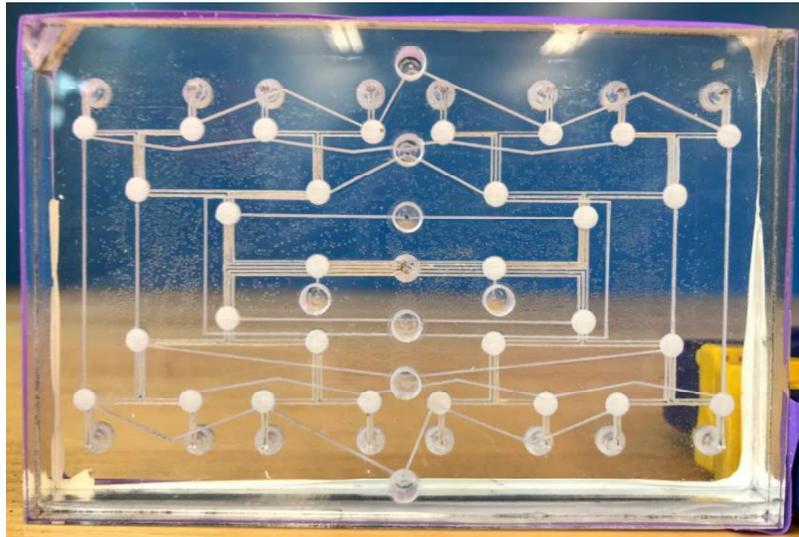


Fig. 21. 4-bit pneumatic demultiplexer chip fabricated using laser engraving and adhesive method.

The resulting pneumatic chip's functionality was tested using the approach outlined in the section 3.2. Controlling the Pneumatic Demultiplexer, i.e., each output was tested by supplying vacuum (digital 1) or atmosphere (digital 0) to its respective select channels. The data channel was supplied with a 70.1 kPa vacuum (digital 1). Given below in Table

and Table are the truth tables showing the test results for the 4-bit pneumatic demultiplexer designed using the branching technique and fabricated using the laser engraving and adhesive technique.

Select Channels (Neg. kPa)				Output Channels (Neg. kPa)							
S0	S1	S2	S3	O0	O1	O2	O3	O4	O5	O6	O7
0.0	72.8	71.8	73.3	0.0	0.1	0.0	0.0	0.0	0.0	0.0	62.7
0.0	72.8	71.8	0.0	0.0	0.2	0.0	0.0	0.0	0.0	61.9	0.0
0.0	72.8	0.0	73.3	0.0	0.1	0.0	0.2	0.0	69.1	0.0	0.1
0.0	72.8	0.0	0.0	0.0	0.0	0.1	0.0	67.4	0.0	0.0	0.0
0.0	72.8	71.8	0.0	0.0	0.1	0.0	69.7	0.0	0.0	0.0	0.0
0.0	0.0	71.8	0.0	0.0	0.0	61.4	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	73.3	0.0	66.2	0.0	0.0	0.0	0.0	0.2	0.0
0.0	0.0	0.0	0.0	66.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0

Table 7. Experimental truth table for the functionality test of the 4-bit pneumatic demultiplexer (outputs O0 to O7) designed using the branching technique and fabricated using the laser engraving and adhesive technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned.

Select Channels (Neg. kPa)				Output Channels (Neg. kPa)							
S0	S1	S2	S3	O8	O9	O10	O11	O12	O13	O14	O15
75.5	72.8	71.8	73.3	0.0	0.0	0.0	0.0	0.1	0.2	0.0	70.5
75.5	72.8	71.8	0.0	0.0	0.1	0.0	0.1	0.0	0.0	62.7	0.0
75.5	72.8	0.0	73.3	0.0	0.0	0.0	0.0	0.0	63.6	0.0	0.0
75.5	72.8	0.0	0.0	0.0	0.2	0.0	0.0	60.5	0.0	0.1	0.0
75.5	0.0	71.8	73.3	0.2	0.0	0.1	72.4	0.1	0.0	0.0	0.1
75.5	0.0	71.8	0.0	0.0	0.0	60.8	0.0	0.0	0.0	0.0	0.0
75.5	0.0	0.0	73.3	0.0	60.8	0.0	0.1	0.0	0.0	0.0	0.0
75.5	0.0	0.0	0.0	64.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 8. Experimental truth table for the functionality test of the 4-bit pneumatic demultiplexer (outputs O8 to O15) designed using the branching technique and fabricated using the laser engraving and adhesive technique. Blue cells: select channels, purple cells: output channels, green cells: outputs functioning as expected, and red cells: outputs that malfunctioned.

The branching technique proved to be successful as all outputs functioned largely as expected. Although, the average pressure drop between data channels and the output

channels increased to 12.5% compared to the 8.9% pressure drop recorded in the 3-bit pneumatic demultiplexer. Although the branching technique is incredibly feasible and easy to implement, it doubles the size of the next iteration pneumatic chip, which has its drawbacks. Since the branching technique does not optimize the number of channels and valves used to produce the next iteration $n+1$ chip, pneumatic demultiplexers developed via this technique are prone to leaks and larger pressure drops. The numerous channels and valves and a larger overall surface area used to produce the $n+1$ pneumatic demultiplexer chip means more areas of weak PMMA-PDMS bonding and a higher possibility of leaks in the chip. A relatively easy potential solution for minimizing the weak bonding areas and leaks is to simply scale down the $n+1$ chips. This would reduce the surface area taken up by the chip which would potentially reduce the risk of more leaks. Pneumatic chips can be scaled down easily but the limit to which the chip can be scaled down depends on the capabilities of the laser cutter being used to fabricate them. In addition, scaling down the chip means the channel widths and depths will also reduce which will have a negative effect on the volumetric flow rate of the pneumatic chip (see previous section 3.4.1. Channel Scaling), hence the scaling limit will also depend on the flow rate requirement.

Chapter 4

4. Evaluation and Demonstration

This chapter describes the tests, experiments, and demonstrations that were performed to characterize the pneumatic demultiplexers designed and fabricated during the course of this research. In addition to the functionality tests (included in subsections 3.3.1. Micromilling and Direct Bonding Method, 3.3.2. Laser Engraving and Adhesive Method, and 3.4.2. Branching) and channel scaling experiments (included in subsection 3.4.1. Channel Scaling), exploration was done to study the latching and non-latching behavior, the pressure decay over time, rise times, frequency response, and bandwidth of the pneumatic demultiplexer. This chapter includes each of the aforementioned explorations, including detailed procedures, experimental results, and discussions of each functionality or metric being investigated.

4.1. Latching Behavior

As discussed in the section 2.2. Monolithic Membrane Valves, pneumatic demultiplexers possess latching functionality (equivalent to the functionality of an electrical D latch circuit) which allows the outputs of the pneumatic chip to hold their state until that state is altered. This section provides a demonstration of this functionality along with an experimental exploration of the latching period or the rate at which the stored pressure decays.

4.1.1. Latching Demonstration

In order to demonstrate the latching functionality, two output channels of the pneumatic demultiplexers were “scoped” (connected) via the I2C pressure sensors and the pressure data were observed and recorded using the generalized controller software (mentioned in the subsection 3.2.2. Generalized Controller Software). To obtain the required data, the select bits to “turn on” (produce vacuum or logical 1 state) the first channel were supplied (using the solenoid valves, see *Driver Hardware Setup*), after a specified amount of time the select bits were changed so that the first channel could latch. The same process was then repeated for the second output channel. This was done so that both outputs would latch at the same time, hence illustrating the latching functionality of the pneumatic demultiplexers.

The experiment described above was performed on the 3-bit pneumatic demultiplexer using outputs O0 and O7. Both outputs O0 and O7 were connected to pressure sensors. Initially select bits were set to 000_2 by keeping S1, S2, and S3 open to the atmosphere (logical 0) and by supplying $\neg S1$, $\neg S2$, and $\neg S3$ with vacuum (logical 1), as explained in the section 3.2. Controlling the Pneumatic Demultiplexer. Once the pressure at output O0 had been constant for a reasonable period of time (~ 25 seconds), the select bits were changed allowing output O0 to be latched. The select bits were then set to 111_2 in order to “turn on” output O7, which was then “turned off” after the pressure at output O7 had been constant for a period of time, thus repeating the process that was performed for output O0. During the whole process, the pressures from O0 and O7 were simultaneously recorded in real-time using the pressure sensors and were plotted versus

time. Fig. 22 shows the resulting pressure-time plot demonstrating the latching functionality.

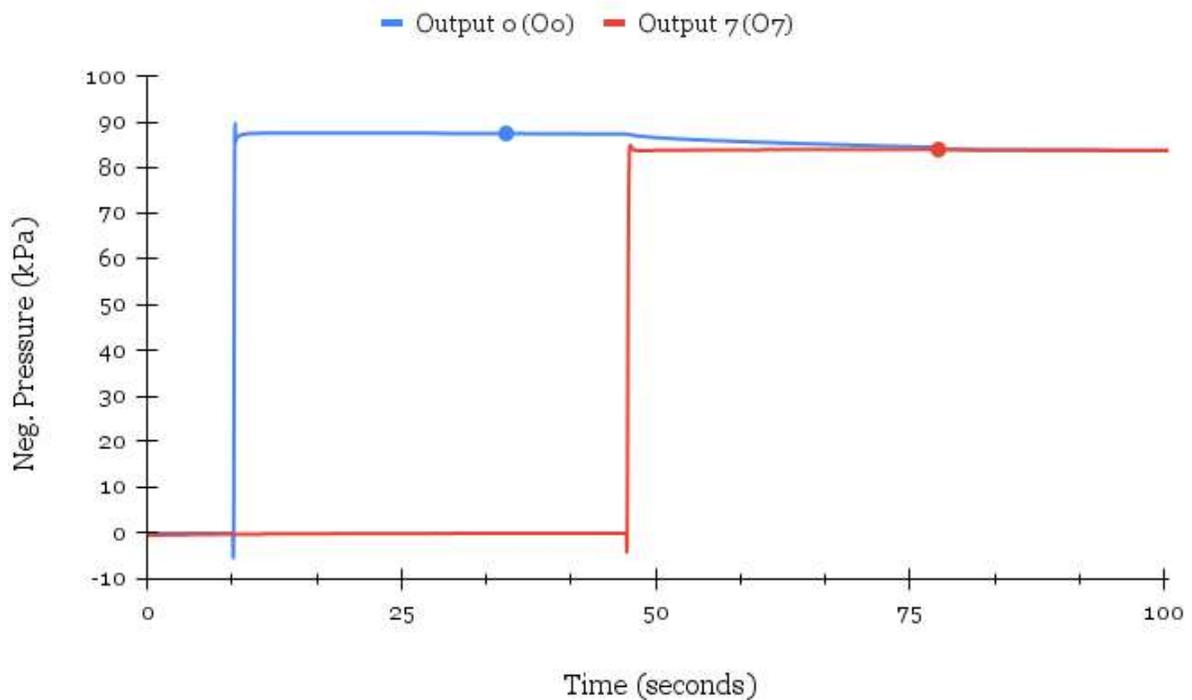


Fig. 22. Pressure against time plot illustrating the latching functionality of the 3-bit pneumatic demultiplexer where the circular markers represent the points at which the outputs were “turned off”.

The experiment mentioned above was repeated for the 4-bit pneumatic demultiplexer with outputs O1 and O14. Fig. 23 shows the resulting pressure-time plot showing the latching functionality for the 4-bit pneumatic chip.

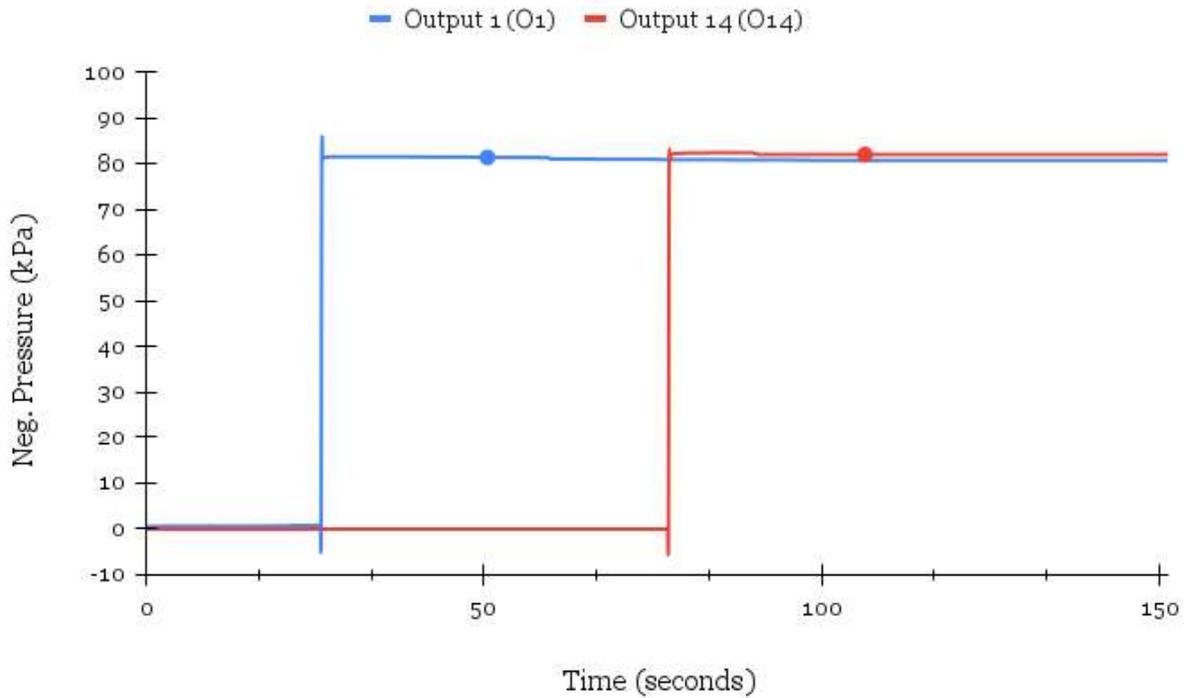


Fig. 23. Pressure against time plot illustrating the latching functionality of the 4-bit pneumatic demultiplexer where the circular markers represent the points at which the outputs were “turned off”.

From plots in Fig. 22 and Fig. 23, it is evident that outputs of the pneumatic demultiplexers can latch both individually and together. Although the outputs maintain the pressure values when latched, there can be a coupling effect when two outputs are latched or “turned on” at the same time. In Fig. 22, the pressure value for output O0 of the 3-bit pneumatic demultiplexer appears to decay at a faster rate when the output O7 is “turned on”. This can be attributed to the increased number of channels and areas that the air can flow through within the “latched” parts of the pneumatic chip,

increasing the possibility of there being areas of weak PMMA-PDMS bonding which would allow trapped air to leak to the surrounding environment.

4.1.2. Latching Period

Theoretically a latched output should retain its state indefinitely; however, in reality due to minuscule leaks in the pneumatic chip, the stored pressure decays over time. The leaks can be due to microscopic irregularities and impurities on the surface of the PMMA wafer which affect the PMMA-PDMS bonding. This section describes the exploration done to find the period of time for which outputs of the pneumatic demultiplexer stay latched. This metric is important for characterizing the pneumatic demultiplexer because soft robotics applications often require multiple actuators to simultaneously hold their state; hence, it is essential to know how long the controlling pneumatic chip can remember the set state. Before investigating the latching periods for each output of the 3-bit and 4-bit pneumatic demultiplexers, it is important to have a specific definition for this metric. Based on the scope of this research, the latching period can be defined as the time it takes for the latched output's pressure to decay to -50 kPa from its stabilized peak pressure. In the definition, the pressure decay value is set to -50 kPa instead of 0 kPa because many small vacuum-operated pneumatic actuators require at least -50 kPa pressure to operate (Adami & Seibel, 2018).

The first step to finding the latching periods is investigating the type of decay the pressure undergoes. To do so, a certain output can be latched by “turning it on” and then allowing it to latch. The pressure from the latched output can be then observed and recorded over a relatively long period of time, the recorded pressure can be plotted

against time and trendlines can be fit to the plot to find a line of best fit and the type of decay the pressure undergoes in latched outputs of the pneumatic demultiplexer. This experiment was performed on output Oo of the 3-bit pneumatic demultiplexer, where output Oo was latched and connected to a pressure sensor. The pressure readings were recorded over an interval of 40 minutes. Fig. 24 shows the resulting pressure-time plot.

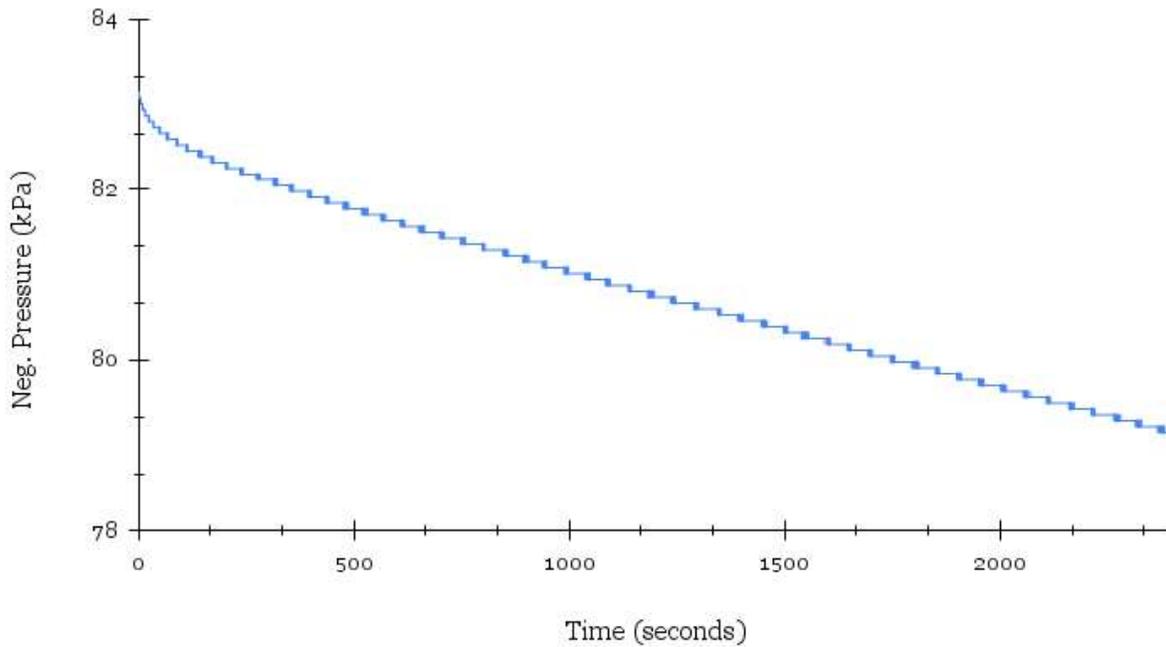


Fig. 24. Pressure against time plot to investigate decay type for long-term pressure decay in latched outputs for output Oo of the 3-bit pneumatic demultiplexer

Given below are the different trendlines applied to the pressure-time curve from Fig. 24 to find the best-fit trendline to characterize the decay.

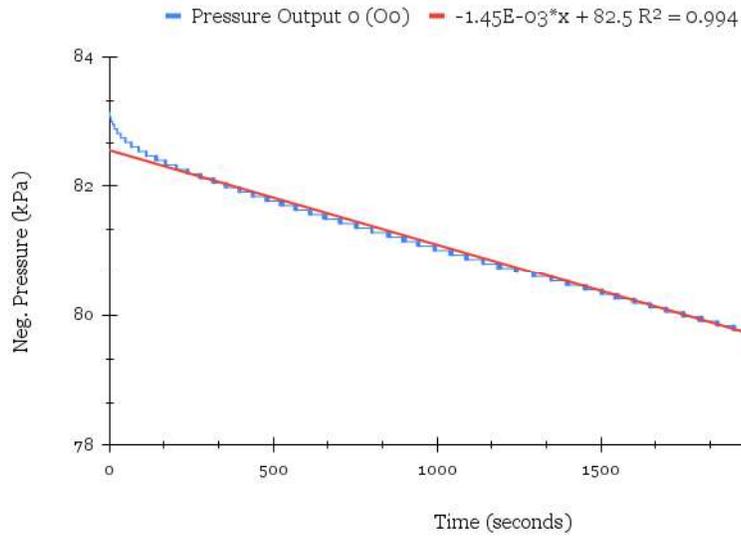


Fig. 25. Pressure against time plot from Fig. 24 with a linear trendline and residual $R^2 = 0.994$

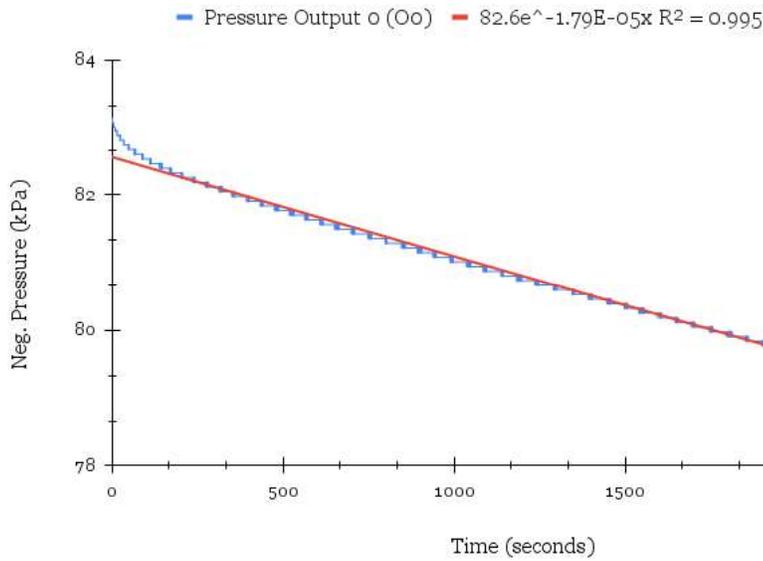


Fig. 26. Pressure against time plot from Fig. 24 with an exponential trendline and residual $R^2 = 0.995$.

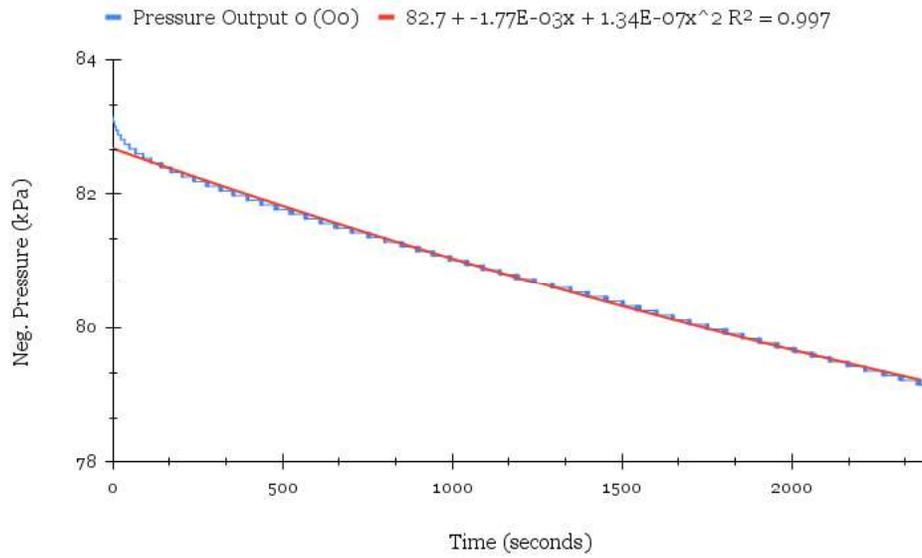


Fig. 27. Pressure against time plot from Fig. 24 with a 2-degree polynomial trendline and residual $R^2 = 0.997$.

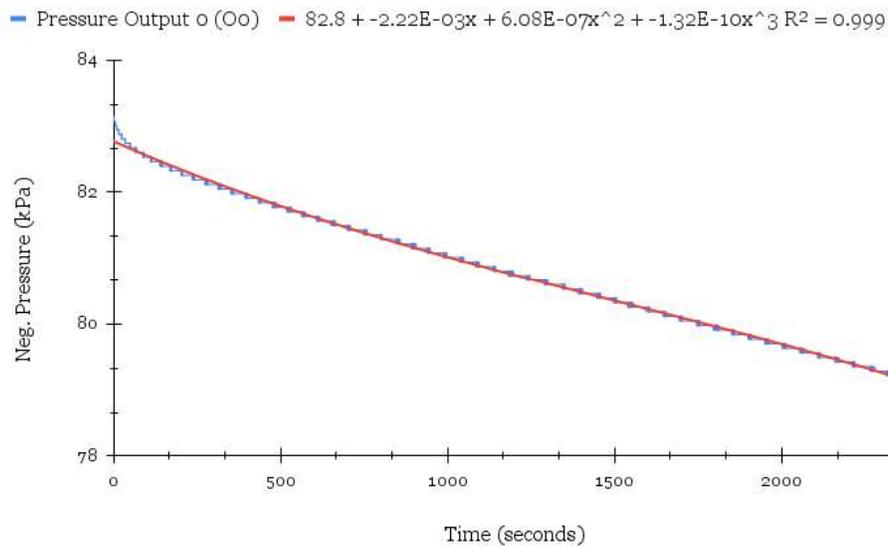


Fig. 28. Pressure against time plot from Fig. 24 with a 3-degree polynomial trendline and residual $R^2 = 0.999$.

From all the plots with trendlines given above, the trendline in Fig. 25 was chosen as the model to characterize the pressure decay in latched outputs of the pneumatic demultiplexer. The chosen trendline is linear with the following equation:

$$y = -0.00145x + 82.5 \tag{8}$$

Based on the long-period pressure-time plot given in Fig. 24 it can be inferred that the pressure decay is mostly linear except for the first ~150-second interval. Although polynomial and exponential models better fit the initial 150 s decay (evident from the higher R^2 residual values) the linear model is a more effective fit for the rest of the decay and is more generalizable. Moreover, most soft robotics applications like grippers and locomotive robots only require latching for at most several minutes. Taking all the aforementioned points into consideration, the linear model was chosen as the best fit for the pressure decay due to its simplicity and general nature.

Now that it has been established that a linear model best fits the pressure decay curve, latching periods for each output in the 3-bit and 4-bit pneumatic demultiplexer can be computed. To do so, outputs are individually latched and connected to a pressure sensor, and the pressure is recorded over a period of approximately 2 minutes. The pressure can then be plotted against time and a linear trendline fit to the pressure-time curve. The equation of the linear can be used to calculate the time for the pressure to decay to -50 kPa. The value obtained from this process is the latching period.

The experiment described above was applied to output Oo of the 4-bit pneumatic demultiplexer. Fig. 29 shows the resulting pressure-time plot with the linear trendline.

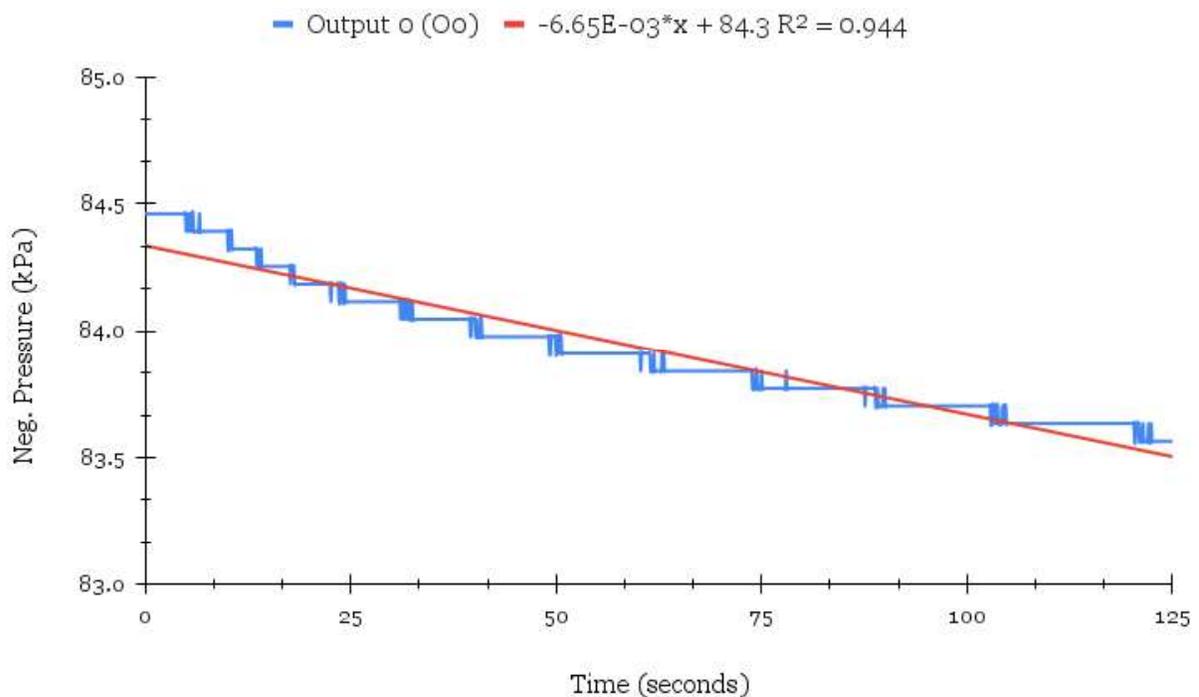


Fig. 29. Pressure against time plot to find the latching period for output Oo of the 4-bit pneumatic demultiplexer.

To determine the latching period the equation of the best-fit line (given below in Equation (9)) can be solved for x where y is 50 kPa. By solving this equation, the value 5157.9 s or 1.4 hours is obtained as the latching period for output Oo of the 4-bit pneumatic demultiplexer.

$$y = -0.00665x + 84.3$$

(9)

The same process was repeated for 4 outputs of the 3-bit and 4-bit pneumatic demultiplexers to obtain more illustrative values for the latching periods for each pneumatic chip. The results for the 3-bit pneumatic demultiplexer are given in Table 7 and the results for the 4-bit pneumatic demultiplexer are given in Table 8.

Output	Latching Period (<i>hrs</i>)
O0	2.9
O3	2.5
O5	3.8
O6	2.3
Average	2.9

Table 7. Latching periods for 3-bit pneumatic demultiplexer.

Output	Latching Period (<i>hrs</i>)
O0	1.4
O6	2
O10	1.4
O14	2.3
Average	1.8

Table 8. Latching periods for 4-bit pneumatic demultiplexer.

From Table 7 and Table 8, it is evident that both 3-bit and 4-bit pneumatic demultiplexers can latch for a couple of hours. The 3-bit pneumatic demultiplexer has a latching period of approximately 2.9 hours whereas the 4-bit pneumatic demultiplexer has a latching period of approximately 1.8 hours. Note that these latching periods may vary depending on the number of other outputs that are latched at the same time because of the coupling effect observed during the latched behavior demonstration (see previous subsection 4.1.1.). However, they prove that the pneumatic demultiplexers implemented and fabricated during the course of this research can latch for a minimum of an hour and hence can support a variety of soft robotic applications.

It is worth noting that the latching period for the 3-bit pneumatic demultiplexer is approximately 1.6 times higher than the latching period for the 4-bit pneumatic demultiplexer. As explained in the subsection 3.4.2. Branching, a potential reason for this can be the larger surface area and the increased number of channels and valves (in

comparison with the 3-bit pneumatic chip) which may cause more leaks. If the 4-bit chip has a higher number of areas from where air can leak, then the higher rate of pressure decay and lower latching period is justifiable.

4.2. Unlatching Behavior

In addition to latched behavior, the pneumatic demultiplexers also exhibit unlatched behavior which can be observed when one or more of the output channels are left open to the atmosphere. When the output channels are vented, the monolithic membrane valves are not able to trap pressure differentials, which means in this state the pneumatic demultiplexer closely resembles the behavior of its electrical counterpart. This means that when an output is “turned on” it will instantaneously output vacuum pressure (logical 1), whereas when the same output is “turned off” its output will change almost instantaneously to atmospheric pressure (logical 0) instead of latching or maintaining that state. This section includes a demonstration of this functionality. Since unlatched behavior resembles the functioning of a digital demultiplexer, digital circuit characterizations and metrics like frequency response and bandwidth are also investigated in this section.

4.2.1. Unlatching Demonstration

The unlatching functionality can be demonstrated simply by “scoping” (connecting) one of the outputs and one of the “scoped” output’s select bit channels. Once both the channel and channels are connected to the I2C pressure sensor, the select bits needed

for the “scoped” output can be supplied, ensuring that the “scoped” select bit is “turned on” (supplied vacuum) last, in order to minimize any unintended delays in the data. After a certain amount of time, the “scoped” select bit can be “turned off” (open to atmosphere). This process can be repeated several times to observe how the input and output pressure pulses.

The experiment described above was performed on the 3-bit pneumatic demultiplexer using output O7 and select bit S2. As mentioned before, both the output O7 and the select bit S2 were connected to pressure sensors. In order to “turn on” output O7, the select bits were set to 111_2 by supplying S0, S1, and S2 with vacuum (logical 1) and by keeping $\neg S_0$, $\neg S_1$, and $\neg S_2$ open to atmosphere (logical 0). Since S2 is the channel connected to the pressure sensor, it was supplied vacuum last to ensure that as soon as negative pressure was input into S2, output O7 will “turn on” (output vacuum). During this process, the pressures from O7 and S2 were recorded using the pressure sensor and the controller software. The pressure data obtained from the experiment is plotted against time in Fig. 30 below.

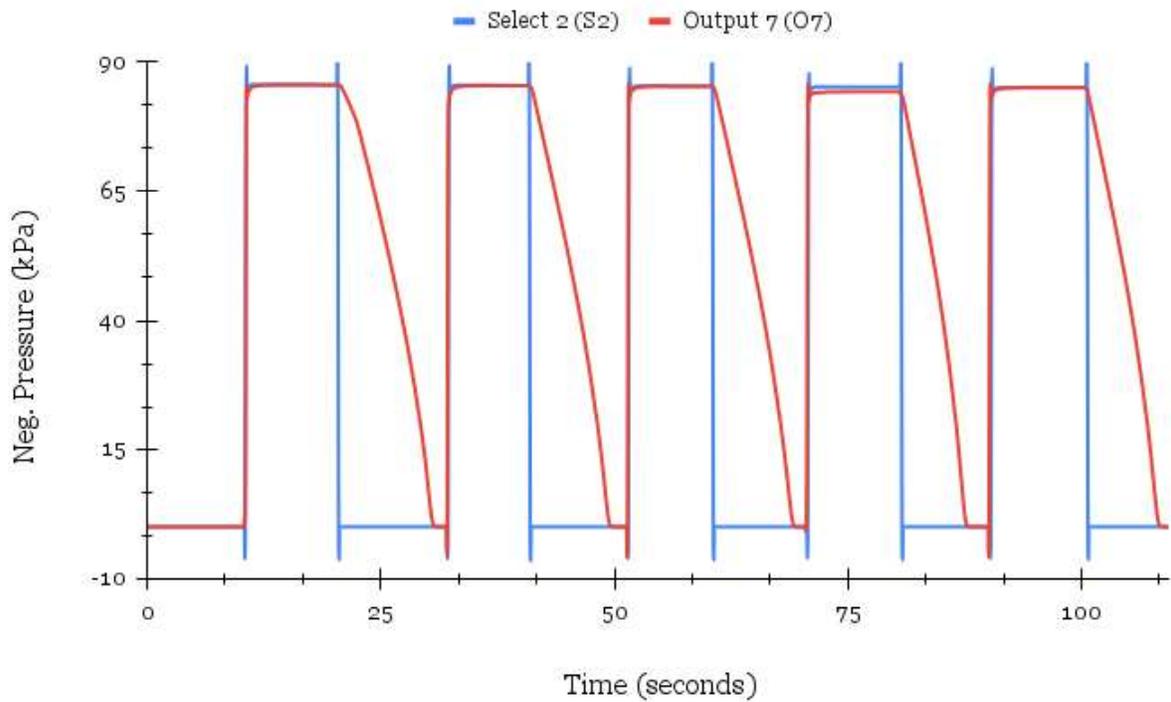


Fig. 30. Pressure against time plot illustrating the unlatching functionality of the 3-bit pneumatic demultiplexer.

The results in this section provide additional proof of the functionality of the pneumatic demultiplexer similar to how an input-output voltage-time plot would provide proof of the functionality of a digital logic circuit. From the plots in Fig. 30, it is evident that the pneumatic demultiplexer functions as a digital demultiplexer. A change in the state of one of the output's select channels causes an immediate change in the respective output. When the select channel is supplied vacuum (logical 1), the output channel instantaneously changes states and outputs vacuum (logical 1) as well. Similarly, when the select channel's state is changed by opening it to the atmosphere (logical 0), the respective output also changes state and the pressure stored in the output begins to

decay. The state change from vacuum (logical 1) to atmosphere (logical 0) does not happen instantaneously because monolithic membrane valves are designed to trap pressure differentials. Even when all the other outputs (except the scoped output) are kept open to the atmosphere it takes the output channels approximately 8 - 10 s to go from vacuum pressures of 78 - 85 kPa to atmospheric pressure of 0 kPa.

4.2.2. Frequency Response and Bandwidth

Now that the unlatching behavior of the pneumatic demultiplexer has been demonstrated, it is important to investigate the frequency response of the pneumatic circuit. Like digital circuits, pneumatic demultiplexers take input signals which are in the form of input pressures. These input pressure signals can be composed of a series of pulses or square waves that can be decomposed into sinusoidal components with differing frequencies. To characterize the pneumatic demultiplexer circuits developed during the course of this project, it is important to study the effect of higher-frequency input signals on the output. As is the case for digit circuits, the pneumatic circuit will attenuate or filter out some of these high-frequency components, resulting in a distorted or degraded signal. Hence, it is important to find the bandwidth of the pneumatic demultiplexer circuits. Bandwidth specifies the usable range of frequencies that pass from input to output. In electronics, this range is often defined relative to the maximum value and is most commonly the point where the signal amplitude is 70.7% of its maximum value (Oppenheim *et al.*, 1997). The same definition can be used in this context, where the bandwidth of the pneumatic demultiplexer circuit can be defined as

the maximum frequency for the input pressure that can be passed to the output without distortion or degradation of the output pressure “signal”.

In order to study the frequency response and find the bandwidth of the pneumatic demultiplexer circuits, a similar process to the one given in the previous subsection 4.1.1. , was used. An output and one of its select bit channels were “scoped” by connecting them to pressure sensors. The rest of the select channels except the “scoped” ones were turned on. After which the solenoid valve for the “scoped” select channel was supplied with a square wave with a varying frequency such that the pressure being supplied to the “scoped” select channel followed the same properties as the varying square wave signal. The input square wave signal was varied in a way that the time period of the wave was reduced after three cycles.

The experiment described above was applied to output O₀ and select channel S₂ of the 3-bit pneumatic demultiplexer. Fig. 31 shows the resulting pressure-time plot with the input and output pressures.

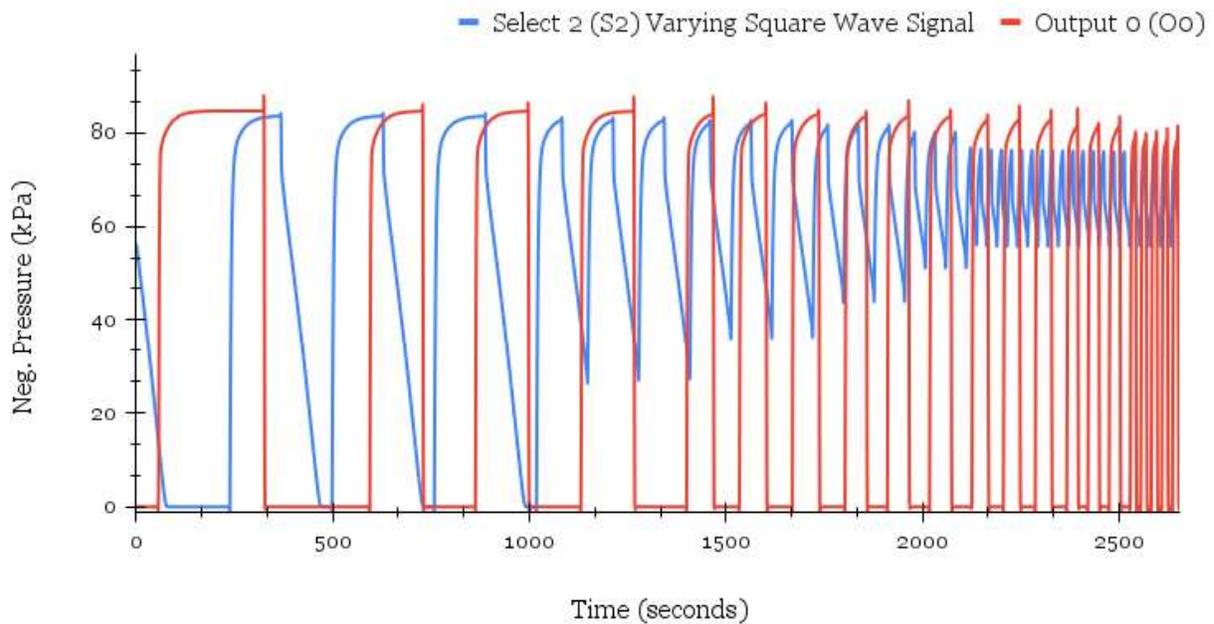


Fig. 31. Pressure against time plot illustrating the frequency response of output Oo of the 3-bit pneumatic demultiplexer with a varying square wave as the input signal.

From the plot in Fig. 31, it can be seen that when a square wave signal of frequency 3.7 Hz is input, the output pressure is not able to reach the atmospheric pressure (logical 0), additionally, the amplitude of the output signal becomes less than 70% of its peak amplitude. Hence the bandwidth of the 3-bit pneumatic demultiplexer is 3.7 Hz.

The same process was applied to output O7 and select channel S1 of the 4-bit pneumatic demultiplexer. Fig. 32 shows the resulting pressure-time plot with the input and output pressures.

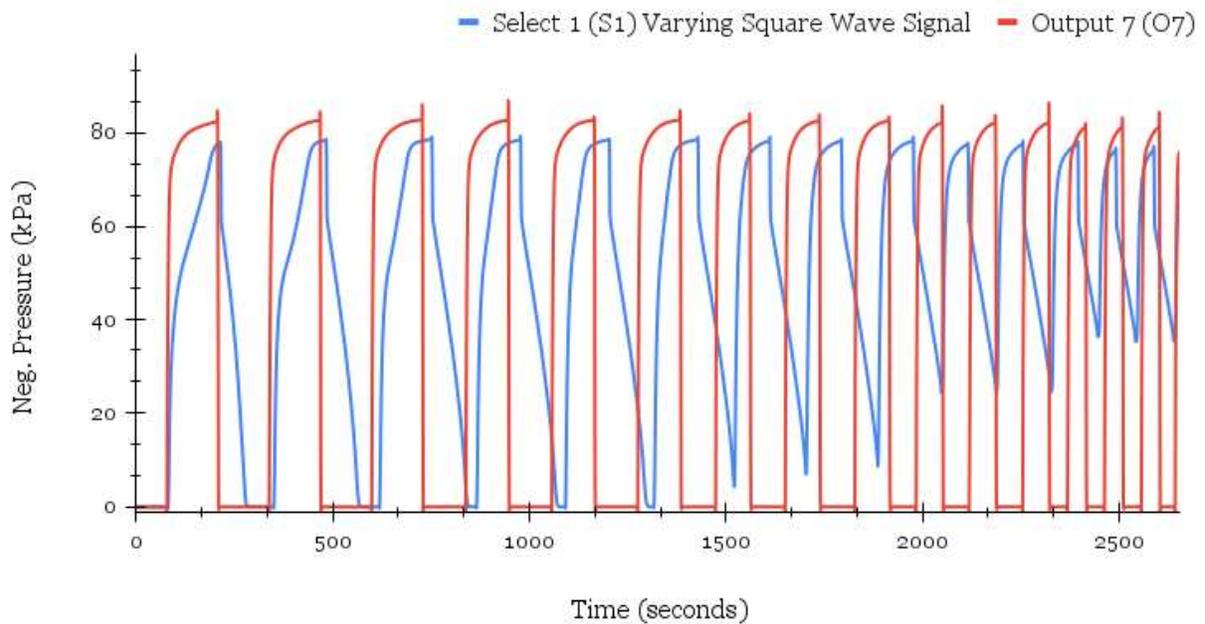


Fig. 32. Pressure against time plot illustrating the frequency response of output O7 of the 4-bit pneumatic demultiplexer with a varying square wave as the input signal.

Similar to the process followed for the 3-bit pneumatic demultiplexer, by analyzing the data of the plot in Fig. 32, it can be seen that when a square wave signal of frequency 3.0 Hz is inputted the output pressure is not able to reach the atmospheric pressure (logical 0); moreover, the amplitude of the output signal becomes less than 70% of its peak amplitude. Hence the bandwidth of the 4-bit pneumatic demultiplexer 3.0 Hz.

4.3. Rise Time

Although the pneumatic demultiplexer is analogous to a digital circuit with two discrete states, vacuum (logical 1) and atmosphere (logical 0), in reality, the pneumatic demultiplexer outputs analog pressure. Moreover, the pneumatic circuit does not instantaneously reach its peak pressure value. In actuality, it takes the pneumatic system a certain amount of time to do so. The time it takes to reach the peak pressure can be defined as the rise time. However, like electrical analog systems, the pneumatic system may never reach its peak value and may indefinitely oscillate about a value less than the peak. Hence it is important to devise a more specific definition for this metric. In electronics, the rise time is defined as the time interval between the instant when a response voltage reaches 10% of its peak value and the instant when it reaches 90% of its peak value (Rumi *et al.*, 2021). The same definition for rise time can be applied to the pneumatic demultiplexer circuit, where the rise time can be described as the time interval between the instant the output pressure reaches 10% of its peak value and the instant it reaches 90% of its peak value.

In order to determine the rise time for outputs in the pneumatic demultiplexer, first the output being investigated must be vented to ensure that it is not latched. This can be accomplished by simply opening the output channel to the atmosphere. Once the output is vented, it should be connected to a pressure sensor and the pressure values should be observed and recorded using the controller software. The output under consideration should be “turned on” by supplying the required select channel configuration (as explained in previous sections in this chapter and section 3.2. Controlling the Pneumatic

Demultiplexer). The pressure should be allowed to stabilize and the pressure data should be plotted against time. Once plotted, the peak value should be computed and the time interval between the pressure value at 10% and 90% of the peak value should be determined to get the required rise time.

The experiment described above was applied to output O0 of the 3-bit pneumatic demultiplexer. Fig. 33 shows the resulting pressure-time plot with red circular markers for the intervals between the pressure values at approximately 10% and 90% of the peak pressure value.

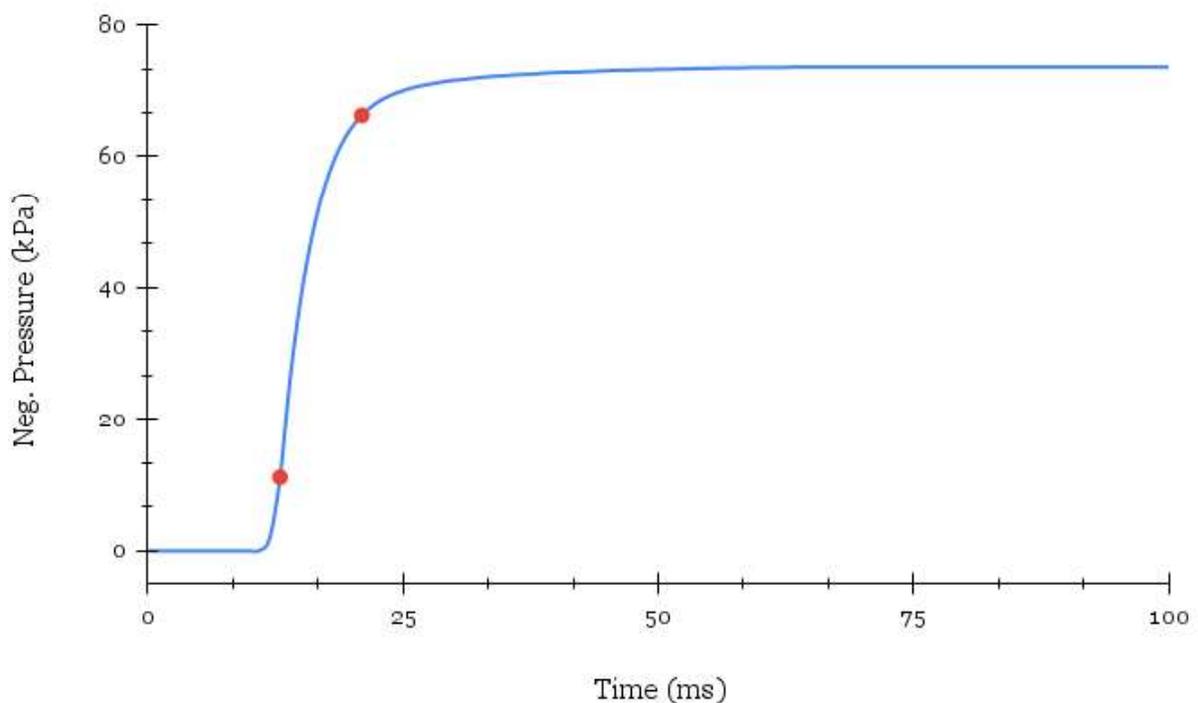


Fig. 33. Pressure against time plot to find rise time for output O0 of the 3-bit pneumatic demultiplexer.

Since the pressure reaches 10% of the peak value at 13 ms and 90% of the peak value at 21 ms as indicated by the markers in Fig. 33, the rise time for output O6 of the 3-bit pneumatic demultiplexer is 8 ms (21 ms - 13 ms). The same process was repeated for 4 outputs of the 3-bit and 4-bit pneumatic demultiplexers to obtain additional illustrative values for the rise times for each pneumatic chip. The results for the 3-bit pneumatic demultiplexer are given in Table 9 and the results for the 4-bit pneumatic demultiplexer are given in Table 10

Output	Rise Time (ms)
O0	9
O3	7
O5	7
O6	7
Average	7.5

Table 9. Rise times for 3-bit pneumatic demultiplexer.

Output	Rise Time (ms)
O0	14
O6	11
O10	12
O14	10
Average	11.7

Table 10. Rise times for 4-bit pneumatic demultiplexer.

Both the 3-bit and the 4-bit pneumatic demultiplexers have relatively short rise times of 7.5 ms and 11.7 ms respectively. The short rise times will enable the pneumatic chips to support a wide range of soft robotics applications where response time is significant.

4.4. System Demonstration

In order to illustrate all the functionalities of the pneumatic demultiplexer, i.e., demultiplexing, latching, and the “digital” behavior, a system demonstration was developed. The demonstration is mainly comprised of an array of 8 LEDs and 8 I2C pressure sensors in addition to the components of the hardware setup described in the subsection 3.2.1. Driver Hardware Setup.

The main goal of the demonstration is to use the pneumatic demultiplexer circuit to drive the 8 LEDs. This is accomplished by connecting each of the outputs of the

pneumatic demultiplexer circuit to one of the I2C pressure sensors. The pressure sensors are connected to an Arduino Leonardo microcontroller, which continually reads the pressure outputs from each of the 8 pressure sensors. In the Arduino code, each of the pressure sensors is paired with an LED; hence, whenever the pressure reading of a pressure sensor is below -50 kPa (threshold from the section 4.1. Latching Behavior) the respective LED is turned on and whenever the pressure sensor reading exceeds the -50 kPa threshold the respective LED is turned off.

Using this simple demonstration, all the functionalities of the pneumatic demultiplexer can be presented. The demultiplexing functionality can be illustrated since only 3 input buttons on the controller software are required to operate all 8 LEDs. Similarly, the latching functionality can be shown since multiple LEDs can be powered at the same time. Lastly, the digital nature of the pneumatic demultiplexer is represented by simply turning the LEDs on and off instead of controlling their brightness. Fig. 34 shows three states of the pneumatic demultiplexer as demonstrated via the array of LEDs. Note that the LEDs are configured from left to right from the most significant bit (MSB) to the least significant bit (LSB)

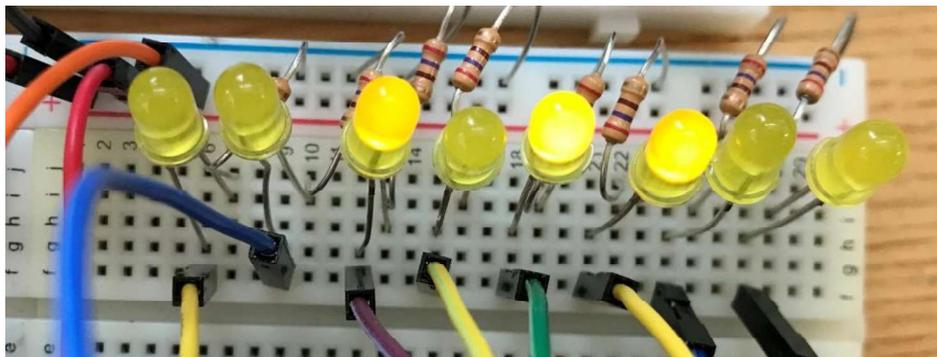
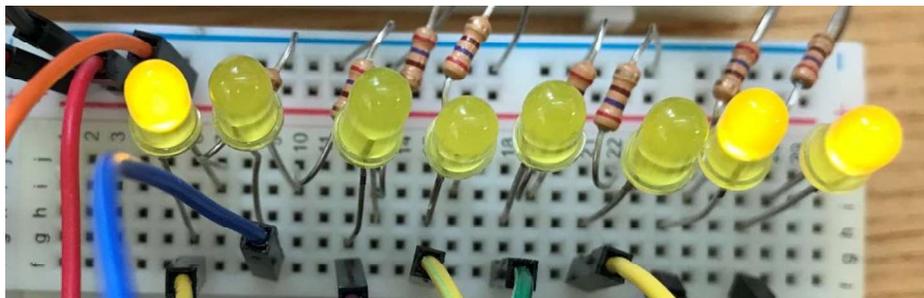
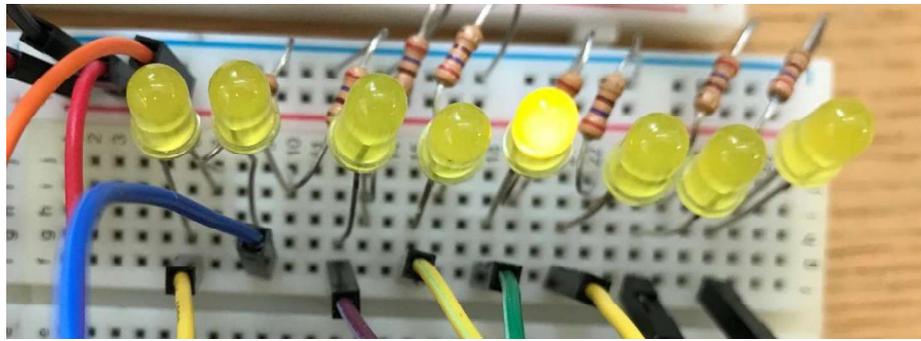


Fig. 34. LED demonstration for the pneumatic demultiplexer circuit showing pneumatic demultiplexer's binary states 00001000_2 , 10000011_2 , and 00101100_2 from top to bottom.

Chapter 5

5. Conclusion

5.1 Discussion

This thesis has successfully addressed the problems with multi-actuator soft robot control mechanisms identified in the section 1.2. Problem by expanding the implementation of pneumatic latched demultiplexer circuits. In this work, 3-bit and 4-bit pneumatic latched demultiplexer circuits were implemented, fabricated, and thoroughly tested. The pneumatic demultiplexers produced during the course of this research have rise times of about 12 ms, latching capability with latching periods of at least 1 hour, and input signal bandwidth of about 3 Hz. This thesis presents an effective method to control multi-actuator soft robots. The demultiplexing functionality of the pneumatic chips allows for the control of 2^n actuators with the use of only n solenoid valves. This makes multi-actuator soft robots significantly easier to build as fewer heavy electromechanical valves and associated rigid electronics are required. Hence this thesis directly addresses one of the major hurdles faced in some of the most promising applications of soft robotics. Although pneumatic demultiplexer circuits for soft robot control previously existed, this research provides a generalized design overview of such circuits and provides an easy scaling solution to design and implement arbitrary n -bit pneumatic demultiplexers. Moreover, previous pneumatic demultiplexer implementations have required relatively complex fabrication methods involving the knowledge and use of CAD software, CNC micromilling, and plasma treatment. The

fabrication method proposed in this research simply requires more readily available and easy-to-use tools and processes like vector design (using common software like Adobe Illustrator or Figma), laser cutting, and double-sided adhesive tape. This makes the manufacturing process of the pneumatic chips considerably more feasible and accessible. Additionally, this thesis outlines the hardware setup needed to control and test multi-actuator soft robotic systems that make use of the pneumatic demultiplexer circuits, as well as provides a generalized controller software that can be used with a variety of different hardware setups, pneumatic chips, and soft robotic devices.

The proposed pneumatic circuits and methods have direct potential applications in various areas discussed in the section 1.1. Motivation. For example, Raeisinezhad *et al.*'s (2020) 81-actuator implementation of the "IntelliPad" could now be powered by a 7-bit pneumatic demultiplexer circuit which can be developed using the technique outlined in this thesis. Tolley *et al.*'s (2014) resilient untethered soft robots could also make use of a pneumatic circuit to reduce the number of rigid elements contained within the body of the soft robot and potentially extend the time over which the robots could be operated untethered. These are two of the many applications of the proposed pneumatic latched demultiplexer circuits in real-world soft robots and soft robotic devices.

5.2 Future Work

Although this work makes significant progress toward addressing some of the problems in soft robot control mechanisms, there are still gaps in research that need to be filled as well as shortcomings to be addressed. Although the laser engraving and adhesive

manufacturing method is feasible and more effective than the conventional micromilling and direct bonding method, the pneumatic chips manufactured using this technique can have areas where the PMMA-PDMS bonding is weak due to problems with adhesive or impurities on the surface of the PMMA. Hence, there is a need to improve this fabrication process, with a specific need to find a more effective cleaning process, as oftentimes impurities settle on the surface of PMMA during the laser cutting process. Another potential area for improvement is the proposed process for creating n -bit pneumatic demultiplexer (see subsection 3.4.2. Branching). Although branching is a simple and easy to implement method, the pneumatic chips developed in this manner significantly increase in size with every iteration and the number of valves and channels increase exponentially. This not only creates a greater potential for leaks but also increases the chance of errors in the design. Hence, there is a need to find ways to optimize the size of the chip along with the number of valves and channels with each iteration. Additionally, there is a need to explore ways to improve the monolithic membrane valves by experimenting with different types of materials and actuation methods to maximize performance. Lastly, a broader area of potential future exploration is the investigation of other types of pneumatic digital logic circuits that can be used to replace electrical circuits commonly used for soft robotics. For instance, pneumatic sensors are worth exploring along with circuits that can process the data from these sensors as this will allow soft robots to react to their environment and external stimuli without the need for microcontrollers. In addition, development of pneumatic memory that can “remember” states of the actuators seems like the immediate next step to the

work done in this thesis and by Huong *et al.* (2021) to “program” soft robots without the need for a tethered electronic circuit.

References

Matthias, B., Oberer-Treitz, S., Staab, H., Schuller, E., & Peldschus, S. (2010, June).

Injury risk quantification for industrial robots in collaborative operation with humans.

In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)* (pp. 1-6). VDE.

Hawkes, E. W., Blumenschein, L. H., Greer, J. D., & Okamura, A. M. (2017). A soft robot that navigates its environment through growth. *Science Robotics*, 2(8), eaan3028.

Blumenschein, L. H., Gan, L. T., Fan, J. A., Okamura, A. M., & Hawkes, E. W. (2018). A tip-extending soft robot enables reconfigurable and deployable antennas. *IEEE Robotics and Automation Letters*, 3(2), 949-956.

Zaidi, S., Maselli, M., Laschi, C., & Cianchetti, M. (2021). Actuation technologies for soft robot grippers and manipulators: A review. *Current Robotics Reports*, 2(3), 355-369.

Tolley, M. T., Shepherd, R. F., Mosadegh, B., Galloway, K. C., Wehner, M., Karpelson, M., & Whitesides, G. M. (2014). A Resilient, Untethered Soft Robot. *Soft Robotics*, 1 (3), 213-223. *Low (50%) Accuracy (%) High (100%)*.

Wang, H., Zhang, R., Chen, W., Wang, X., & Pfeifer, R. (2017). A cable-driven soft robot surgical system for cardiothoracic endoscopic surgery: preclinical tests in animals.

Surgical endoscopy, 31, 3152-3158.

Raeisinezhad, M., Pagliocca, N., Koohbor, B., & Trkov, M. (2020). Intellipad: intelligent soft robotic pad for pressure injury prevention. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 685-690). IEEE.

Iida, F., & Laschi, C. (2011). Soft robotics: Challenges and perspectives. *Procedia Computer Science*, 7, 99-102.

Whitesides, G. M. (2006). The origins and the future of microfluidics. *Nature*, 442(7101), 368-373.

Zhang, Q., Zhang, M., Djeghlaf, L., Bataille, J., Gamby, J., Haghiri-Gosnet, A. M., & Pallandre, A. (2017). Logic digital fluidic in miniaturized functional devices: Perspective to the next generation of microfluidic lab-on-chips. *Electrophoresis*, 38(7), 953-976.

Jensen, E. C., Grover, W. H., & Mathies, R. A. (2007). Micropneumatic digital logic structures for integrated microdevice computation and control. *Journal of Microelectromechanical Systems*, 16(6), 1378-1385.

Xu, K., & Pérez-Arancibia, N. O. (2020). Electronics-free logic circuits for localized feedback control of multi-actuator soft robots. *IEEE Robotics and Automation Letters*, 5(3), 3990-3997.

Preston, D. J., Rothmund, P., Jiang, H. J., Nemitz, M. P., Rawson, J., Suo, Z., & Whitesides, G. M. (2019). Digital logic for soft devices. *Proceedings of the National Academy of Sciences*, 116(16), 7750-7759.

- Grover, W. H., Skelley, A. M., Liu, C. N., Lagally, E. T., & Mathies, R. A. (2003). Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices. *Sensors and Actuators B: Chemical*, 89(3), 315-323.
- Perdigones, F. A., Luque, A., & Quero, J. M. (2014). Correspondence between electronics and fluids in MEMS: Designing microfluidic systems using electronics. *IEEE Industrial Electronics Magazine*, 8(4), 6-17.
- Tokheim, R. (2017). Digital electronics: Principles and applications (9th ed.). *McGraw-Hill Education*, 97-101.
- Scott, S. M., & Ali, Z. (2021). Fabrication methods for microfluidic devices: An overview. *Micromachines*, 12(3), 319.
- Tan, H. Y., Loke, W. K., & Nguyen, N. T. (2010). A reliable method for bonding polydimethylsiloxane (PDMS) to polymethylmethacrylate (PMMA) and its application in micropumps. *Sensors and Actuators B: Chemical*, 151(1), 133-139.
- Gillespie, B., personal communication, 2023.
- Gallardo Hevia, E., McCann, C. M., Bell, M., Hyun, N. S. P., Majidi, C., Bertoldi, K., & Wood, R. J. (2022). High-Gain Microfluidic Amplifiers: The Bridge between Microfluidic Controllers and Fluidic Soft Actuators. *Advanced Intelligent Systems*, 4(10), 2200122.
- Adami, M., & Seibel, A. (2018). On-board pneumatic pressure generation methods for soft robotics applications. *Actuators* (Vol. 8, No. 1, p. 2). MDPI.

Oppenheim, A. V., Willsky, A. S., Nawab, S. H., & Ding, J. J. (1997). Signals and systems (Vol. 2, pp. 74-102). Upper Saddle River, NJ: Prentice hall.

Rumi, A., Marinelli, J. G., Pastura, M., Barater, D., & Cavallini, A. (2021). Insights into the Definition of Converter Surge Rise Time and its Influence on Turn/Turn Electrical Stress. *IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)* (pp. 272-276). IEEE.

Appendix

A. Controller Program

Given below is the `app.py` Python script which generates the user interface and communicates with the microcontroller, as discussed and shown in the subsection 3.2.2.

Generalized Controller Software.

```
import tkinter as tk
from tkinter import ttk
from serial import Serial
import time

from valve import Valve

import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

VALVE_COUNT = 4
BTN_PADX, BTN_PADY = 5, 0
FRAME_PADDING = (20, 10)
VALVE_NAMES = [
    "Input 0", "Input 1", "Input 2", "Input 3"
]

class App(ttk.Frame):
    def __init__(self, parent):
        ttk.Frame.__init__(self)

        self.serial_device = None
        self.valve_arr = [None for _ in range(VALVE_COUNT)]

        self.dstart = False
        self.tcount = -1
        self.ty, self.px = list(), list()

    def setupWidgets(self):
```

```

self.setupInputBtns()
self.setupOutputWidget()
self.setupDataCapture()

def startSerialComm(self, port, baud, timeout):
    try:
        self.serial_device = Serial(port, baud, timeout=timeout)
        self.updateStatus(connected=True)
    except:
        self.updateStatus(connected=False)
        raise ConnectionError("Failed to connect to serial device.")

def exit(self):
    if self.dstart:
        self.dstart = False
        self.serial_device.write(bytes('E', 'utf-8'))
    self.destroy()

def animatePlot(self, i):
    vrcv = False

    if not self.dstart:
        self.serial_device.write(bytes('S', 'utf-8'))
        self.dstart = True

    if not self.ani:
        raise ValueError("Plot animation never started.")

    try:
        rval = self.serial_device.readline().decode()
        vrcv = True
        self.updateStatus(connected=True)
    except:
        vrcv = False
        self.updateStatus(connected=False)
        print("Serial device not responding.")

    try:
        pval = float(rval.split(':')[1])*-1.0
        # pval = float(rval.split(':')[1])
        vrcv = True
        self.updateStatus(connected=True)
    except:
        vrcv = False
        self.updateStatus(connected=False)

```

```

        print("Serial device sending unexpected data: {}".format(rval))

    if vrecv:
        self.updateOutput(pvalue=pval)

        self.tcount = self.tcount + 1
        self.ty.append(self.tcount)
        self.px.append(pval)

        ax = plt.gcf().get_axes()[0]
        ax.cla()

        if self.tcount <= self.value_count:
            xmin, xmax = 0, self.value_count
        else:
            xmin, xmax = self.tcount - (self.value_count - 1), self.tcount
            self.px.pop(0)
            self.ty.pop(0)
        ax.set(xlim=(xmin, xmax), ylim=(self.ymin, self.ymax))
        ax.plot(self.ty, self.px)

        ax.set_title("Pressure-Time Scope")
        ax.set_ylabel("Neg. Pressure (PSI)")
        ax.set_xlabel("Time (s)")

def setupPlot(self, value_count, ymin, ymax):
    self.value_count = value_count
    self.ymin = ymin
    self.ymax = ymax

    canvas = FigureCanvasTkAgg(plt.gcf(), master=self)
    canvas.get_tk_widget().grid(column=5, row=0, rowspan=5, padx=5, pady=5)

    plt.style.use('ggplot')

    plt.gcf().subplots(1, 1)

    plt.title("Pressure-Time Scope")
    plt.ylabel("Neg. Pressure (PSI)")
    plt.xlabel("Time (s)")
    plt.axis([0, self.value_count, self.ymin, self.ymax])

    self.ani = FuncAnimation(plt.gcf(), self.animatePlot, interval=1000,
blit=False)

```

```

def setupDataCapture(self):
    self.data_frame = ttk.LabelFrame(self, text="Data Capture",
padding=FRAME_PADDING)
    self.data_frame.grid(column=0, row=3, padx=FRAME_PADDING,
pady=FRAME_PADDING, sticky="nsew")

    self.data_btn = ttk.Button(self.data_frame, text="Start Capture",
style="Accent.TButton")
    self.data_btn.grid(column=0, row=3, sticky="nsew", padx=BTN_PADX,
pady=BTN_PADY)

    self.reset_btn = ttk.Button(self.data_frame, text="Reset Capture",
style="Accent.TButton")
    self.reset_btn.grid(column=1, row=3, sticky="nsew", padx=BTN_PADX,
pady=BTN_PADY)

    self.dwnld_btn = ttk.Button(self.data_frame, text="Download Data .csv",
style="Accent.TButton")
    self.dwnld_btn.grid(column=2, row=3, sticky="nsew", padx=BTN_PADX,
pady=BTN_PADY)

    self.data_lbl = ttk.Label(self.data_frame, text="Not capturing packets
- 0 total data points captured.")
    self.data_lbl.grid(column=0, row=4, colspan=4, sticky="nsew",
padx=BTN_PADX, pady=(BTN_PADY + 10, 0))

def setupOutputWidget(self):
    self.chnll_frame = ttk.LabelFrame(self, text="Output Values",
padding=FRAME_PADDING)
    self.chnll_frame.grid(column=0, row=2, padx=FRAME_PADDING,
pady=FRAME_PADDING, sticky="nsew")

    self.chnll_lbl = ttk.Label(self.chnll_frame, text="Channel 1: ")
    self.chnll_lbl.grid(column=0, row=2, colspan=2, sticky="nsew")

    self.out_lbl = ttk.Label(self.chnll_frame, text="0.0 Neg. PSIG",
background="#1c1c1c", padding=(5, 5), borderwidth=5)
    self.out_lbl.grid(column=3, row=2, colspan=4, sticky="nsew")

def setupInputBtns(self):
    self.input_frame = ttk.LabelFrame(self, text="Input Valves",
padding=FRAME_PADDING)
    self.input_frame.grid(column=0, row=1, padx=FRAME_PADDING,

```

```

pady=FRAME_PADDING, sticky="nsew")

    for vid in range(VALVE_COUNT):
        cstate = tk.IntVar()
        if not self.serial_device:
            raise ValueError("Serial communication never started.")
        cvalve = Valve(vid, cstate, self.serial_device)
        self.valve_arr[vid] = cvalve
        cbtn = ttk.Checkbutton(
            self.input_frame, text=VALVE_NAMES[vid],
style="Toggle.TButton",
            variable = cstate, onvalue = 0, offvalue = 1,
command=cvalve.update
        )
        cbtn.grid(column=vid, row=1, padx=BTN_PADX, pady=BTN_PADY)

    def setupStatusWidget(self):
        self.status_frame = ttk.LabelFrame(self, text="System Status",
padding=FRAME_PADDING)
        self.status_frame.grid(column=0, row=0, padx=FRAME_PADDING,
pady=FRAME_PADDING, sticky="nsew")

        self.serial_lbl = ttk.Label(self.status_frame, text="Serial Device:")
        self.serial_lbl.grid(column=0, row=0, colspan=2, sticky="nsew")

        self.value_lbl = ttk.Label(self.status_frame, text="Connecting...")
        self.value_lbl.grid(column=2, row=0, colspan=2, sticky="nsew")

    def updateStatus(self, connected):
        ONLINE_COLOR = "#1effa3"
        OFFLINE_COLOR = "#ec2227"
        # if online
        if connected:
            self.value_lbl.config(text="Online", foreground=ONLINE_COLOR)
        else:
            self.value_lbl.config(text="Offline", foreground=OFFLINE_COLOR)

    def updateOutput(self, pvalue):
        self.out_lbl.config(text="{: .2f} Neg. PSIG".format(pvalue))

    def updateDataCaptureLbl(self, capturing, dcount):
        cstr = "Capturing" if capturing else "Not Capturing"
        self.data_lbl.config(text="{ } packets - { } total data points
captured.".format(cstr, dcount))

```

```

if __name__ == "__main__":
    root = tk.Tk()
    root.title("Pneumatic Driver")

    root.tk.call("source", "azure.tcl")
    root.tk.call("set_theme", "dark")

    app = App(root)
    app.setupStatusWidget()
    app.startSerialComm(port="COM11", baud=9800, timeout=1)
    app.setupWidgets()
    app.setupPlot(value_count=50, ymin=-0.25, ymax=15)

    app.pack(fill="both", expand=True)

    root.update()
    root.minsize(root.winfo_width(), root.winfo_height())
    x_cordinate = int((root.winfo_screenwidth() / 2) - (root.winfo_width() /
2))
    y_cordinate = int((root.winfo_screenheight() / 2) - (root.winfo_height() /
2))
    root.geometry("+{}+{}".format(x_cordinate, y_cordinate-20))

    root.mainloop()

```

The Valve object used in app.py is encapsulated in its own script valve.py which is given below:

```

class Valve:
    STATE0_OFFSET = 5 # serial message offset to set
state (0)

    def __init__(self, bin_val, state, serial_device):
        self.bin_val = bin_val # binary value of the valve
        self.state = state # state of the valve (0 or 1)
        self.serial_device = serial_device # serial object for controller
device

        self.update()

```

```
def update(self):
    if self.state.get() == 1:
        oval = bytes(str(self.bin_val + 1), 'utf-8')
    elif self.state.get() == 0:
        oval = bytes(str(self.bin_val + 1 + Valve.STATE0_OFFSET), 'utf-8')
    else:
        raise ValueError("Unexpected valve state.")
    try:
        self.serial_device.write(oval)
    except:
        raise ConnectionError("Failed to update valve value.")
```

B. Laser Cutter Settings

The laser cutter settings used to fabricate the pneumatic demultiplexer chips using the *Laser Engraving and Adhesive Method* are given below (the settings may vary based on the laser cutter being used):

Setting Type	Value Used
Resolution	1200 DPI
Job Type	Combined (raster and vector)
Raster Speed	50%
Raster Power	100%
Raster Frequency	50%
Vector Speed	80%
Vector Power	60%
Vector Frequency	100%

Table 11. Laser cutter settings to fabricate pneumatic demultiplexer chips using laser engraving and adhesive method.

Note that the settings shown in the table above (Table 11) were used solely to etch channels and valves into the surface of the PMMA wafer. To cut through holes, the laser cutter's recommended vector settings were used for a 9mm acrylic wafer.

C. Latching Period Plots

Given below are all the latching plots for the 3-bit and 4-bit pneumatic demultiplexers used to find the latching period values given in the subsection 4.1.2. Latching Period.

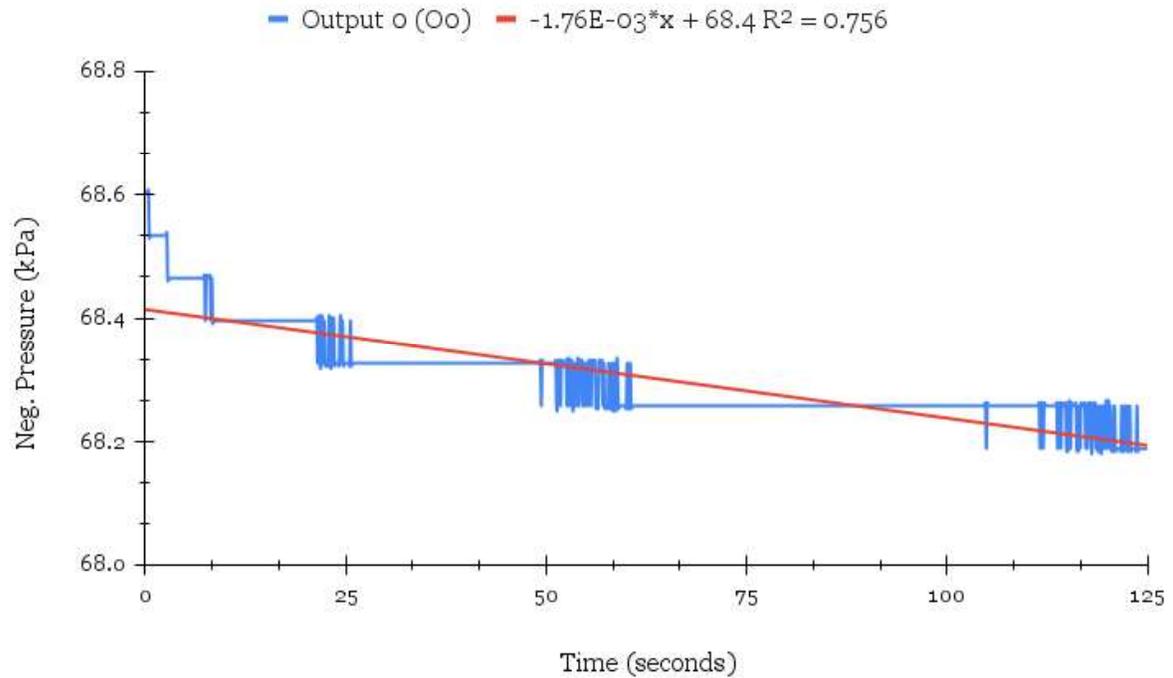


Fig. 35. Pressure against time plot to find the latching period for output Oo of the 3-bit pneumatic demultiplexer.

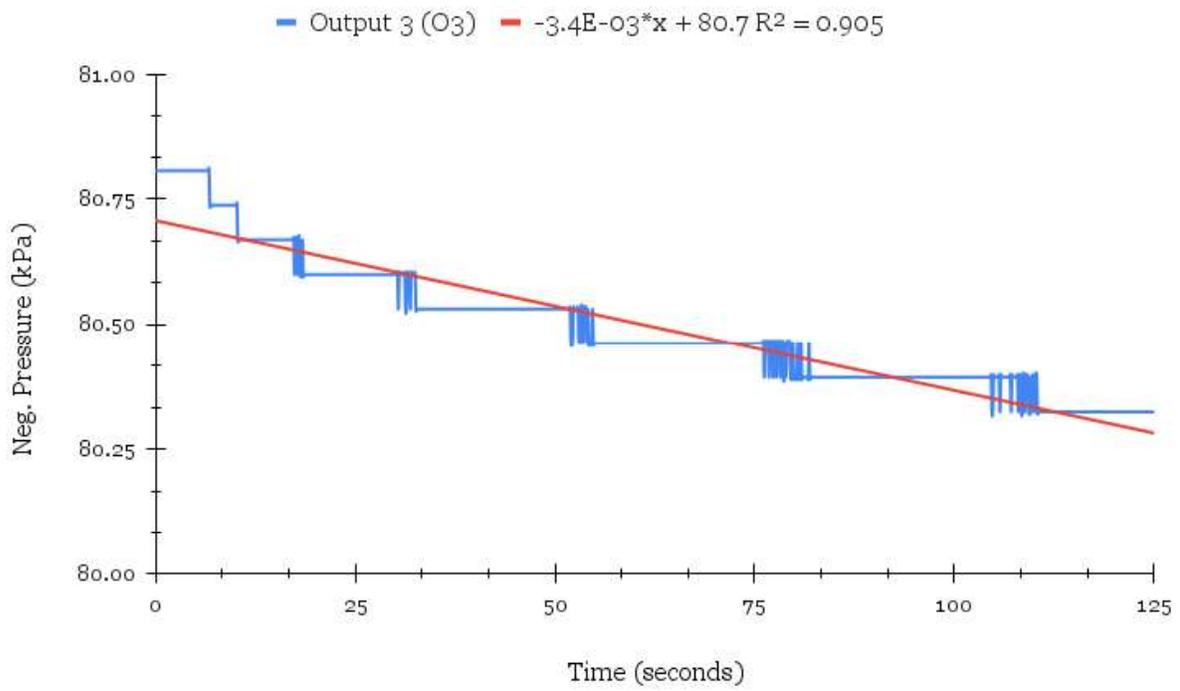


Fig. 36. Pressure against time plot to find the latching period for output O3 of the 3-bit pneumatic demultiplexer.

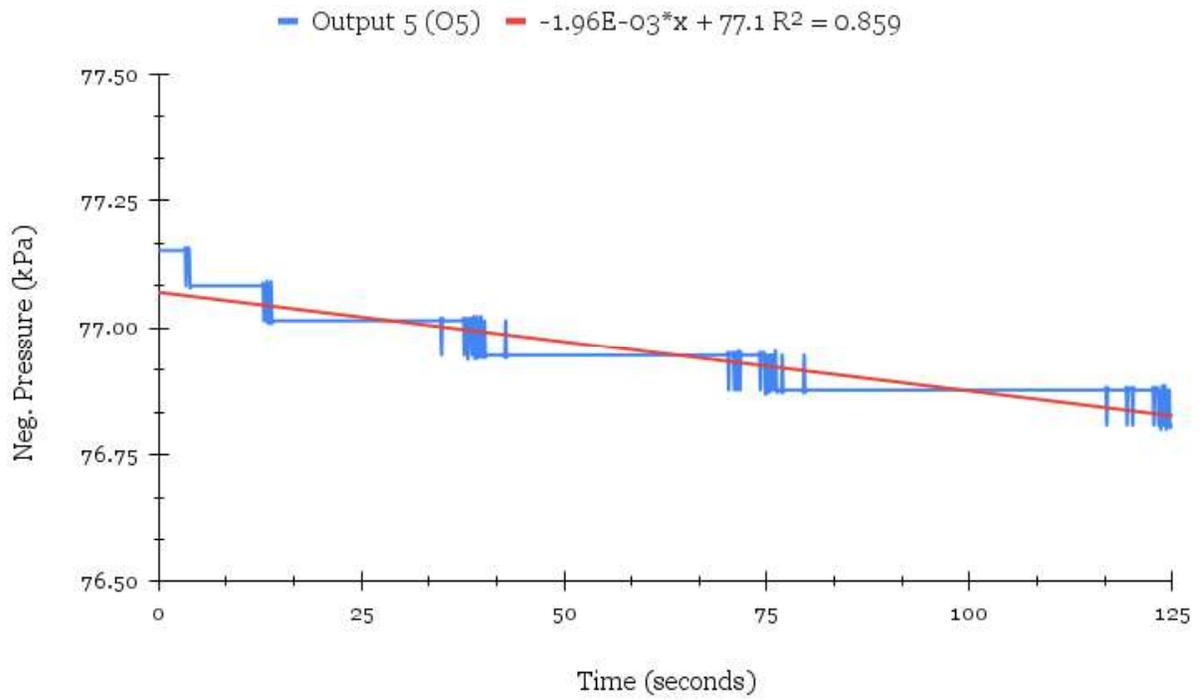


Fig. 37. Pressure against time plot to find the latching period for output O5 of the 3-bit pneumatic demultiplexer.

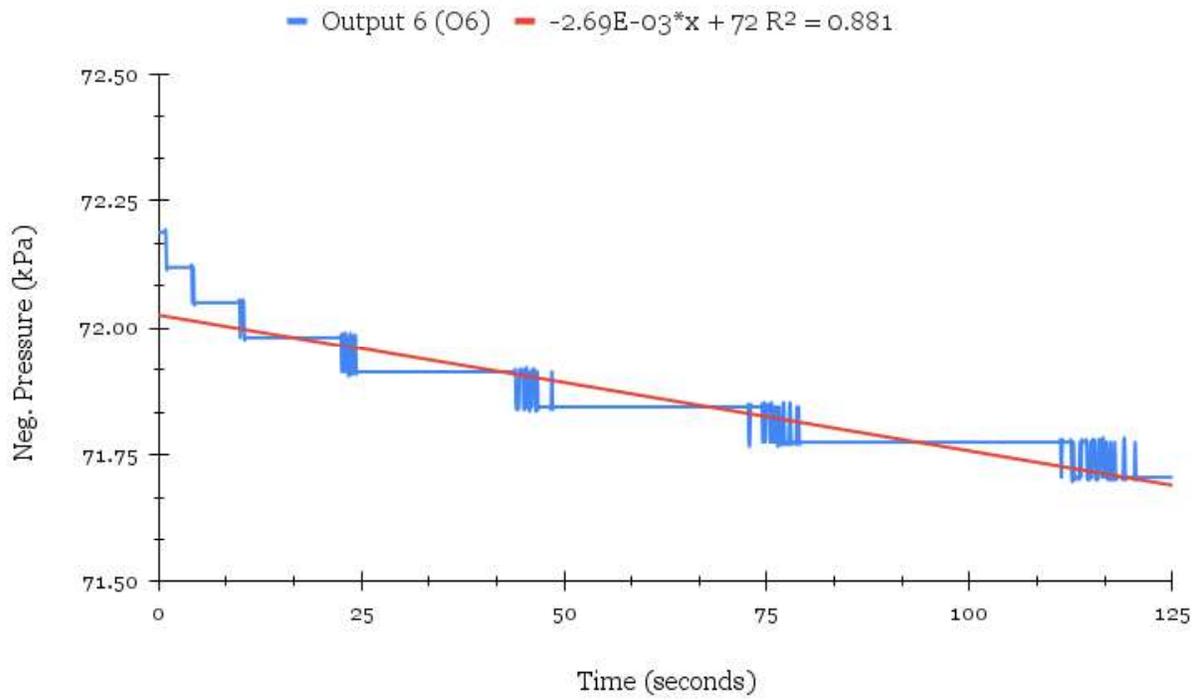


Fig. 38. Pressure against time plot to find the latching period for output O6 of the 4-bit pneumatic demultiplexer.

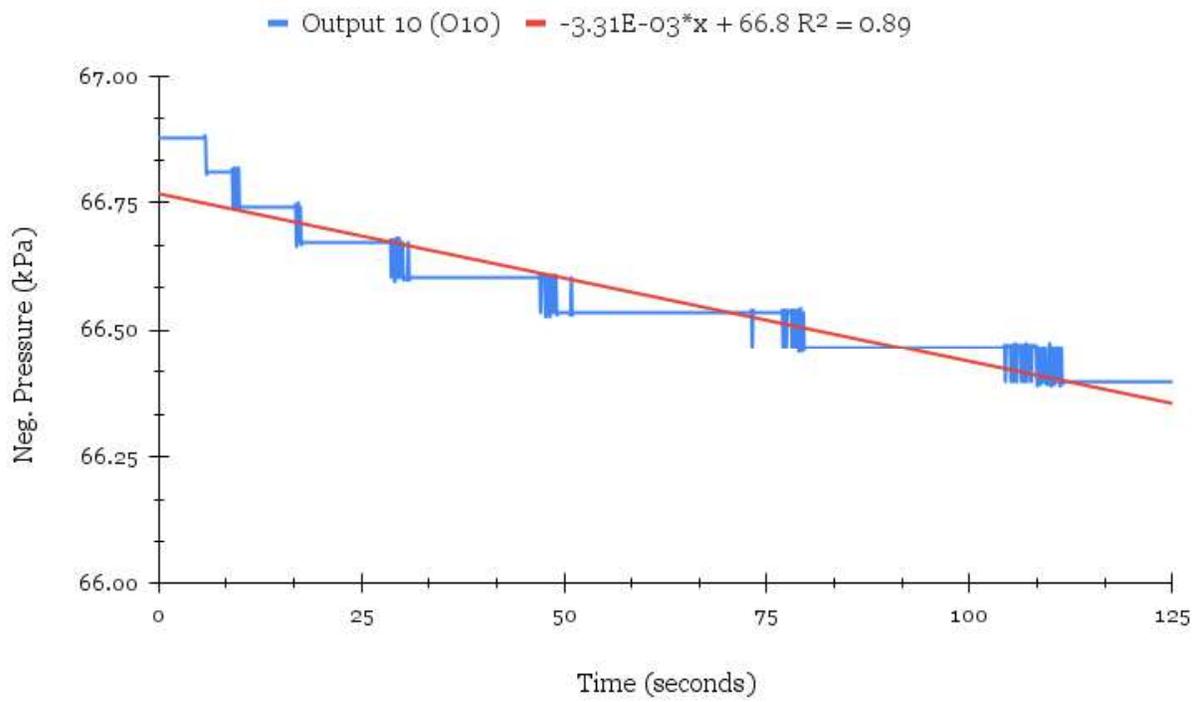


Fig. 39. Pressure against time plot to find the latching period for output O10 of the 4-bit pneumatic demultiplexer.

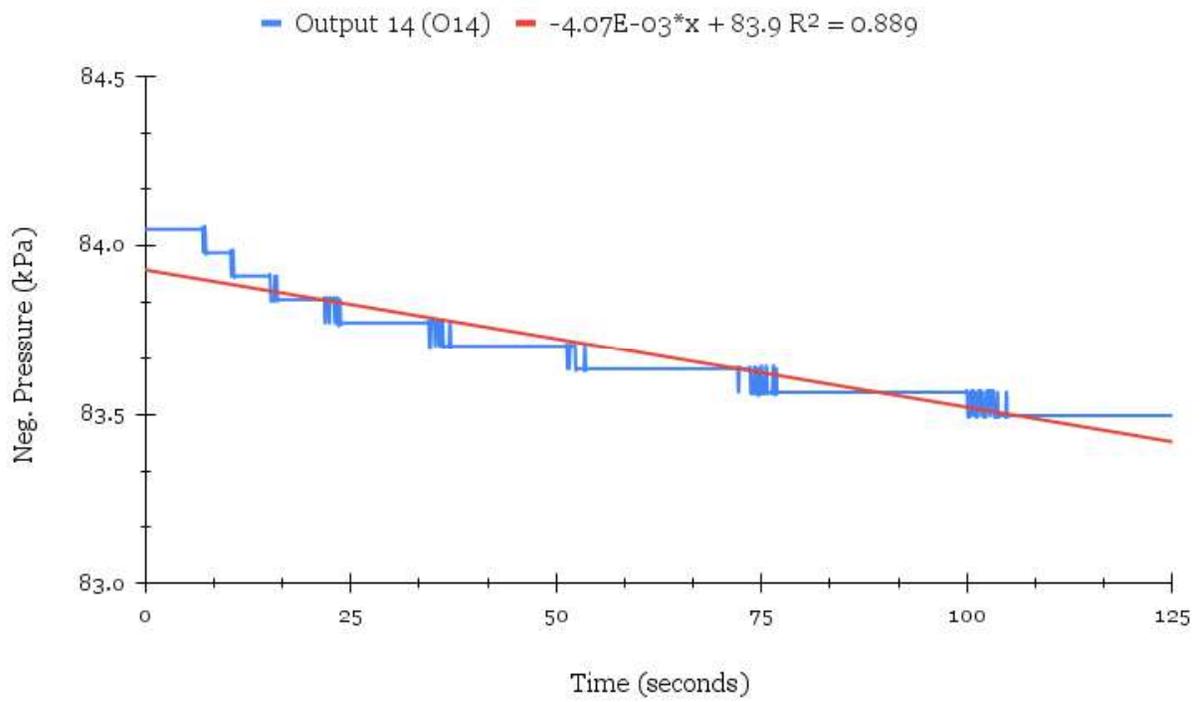


Fig. 40. Pressure against time plot to find the latching period for output O14 of the 4-bit pneumatic demultiplexer.

D. Rise Time Plots

Given below are all the rise time plots for the 3-bit and 4-bit pneumatic demultiplexers used to find all the rise time values given in the section 4.3. Rise Time.

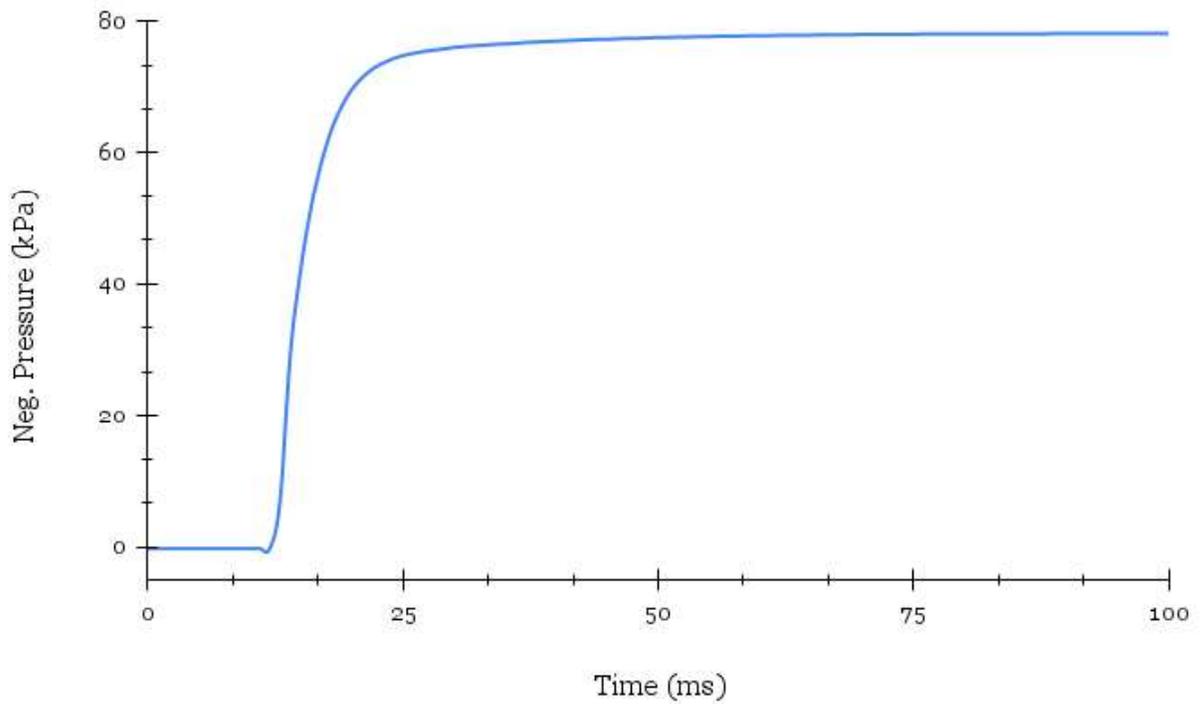


Fig. 41. Pressure against time plot to find rise time for output O3 of the 3-bit pneumatic demultiplexer.

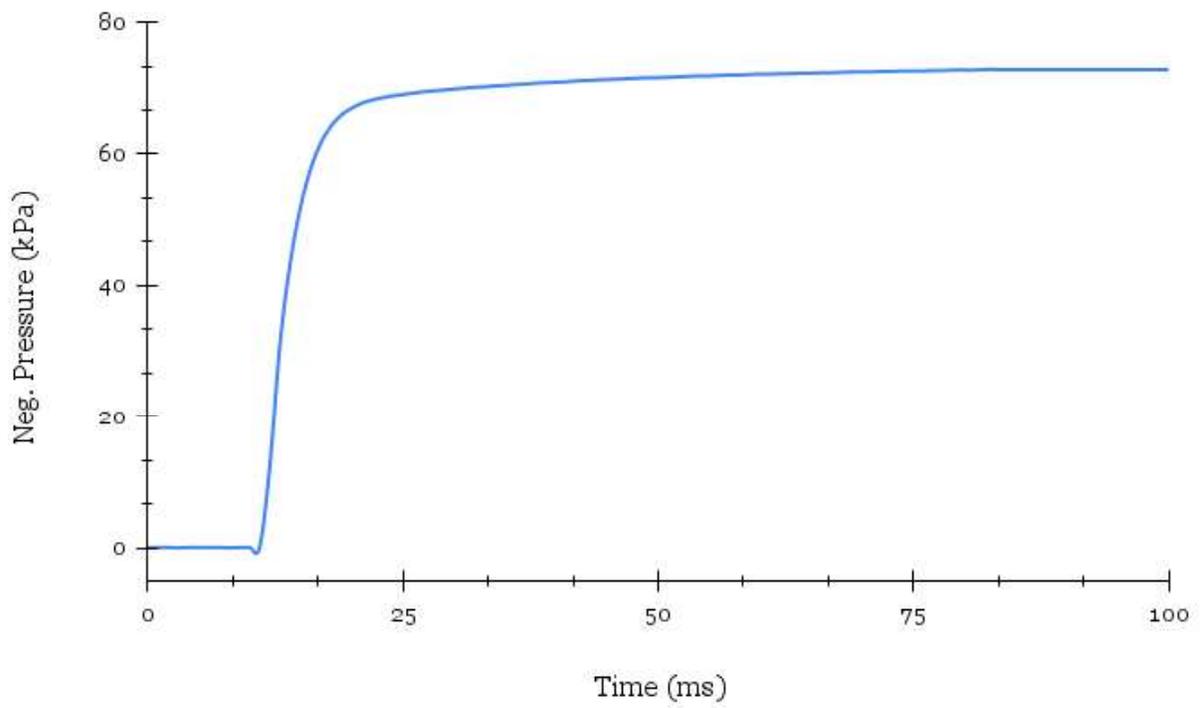


Fig. 42. Pressure against time plot to find rise time for output O5 of the 3-bit pneumatic demultiplexer.

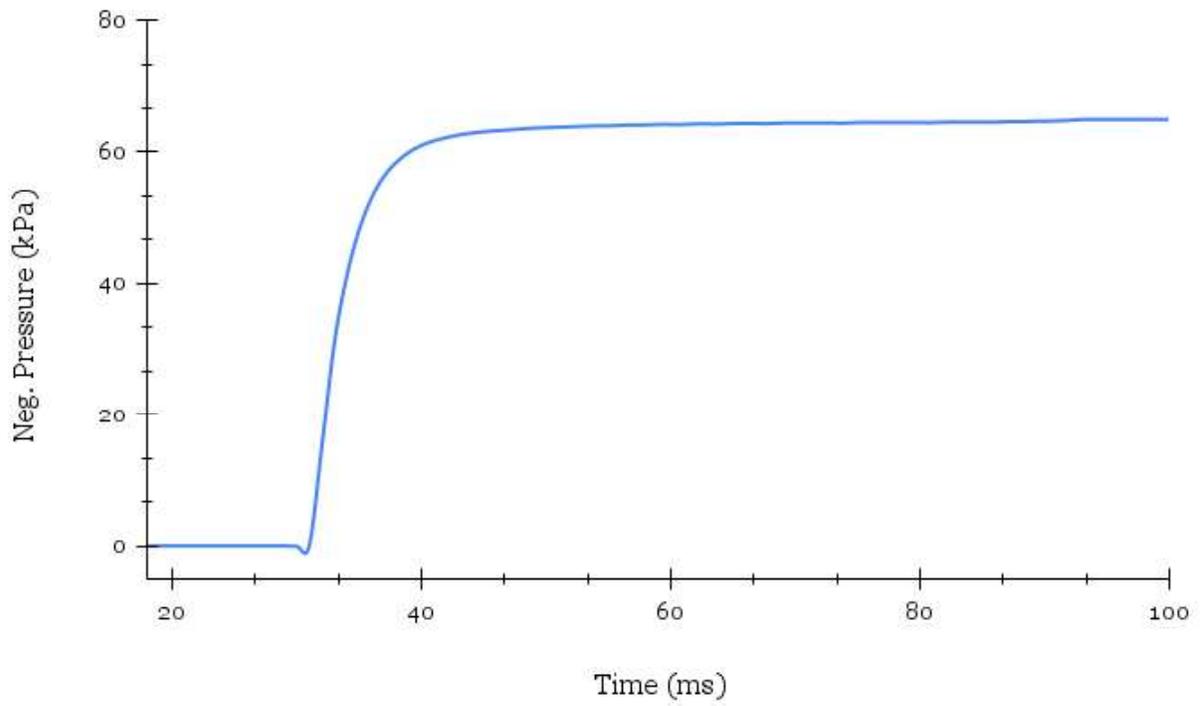


Fig. 43. Pressure against time plot to find rise time for output O6 of the 3-bit pneumatic demultiplexer.

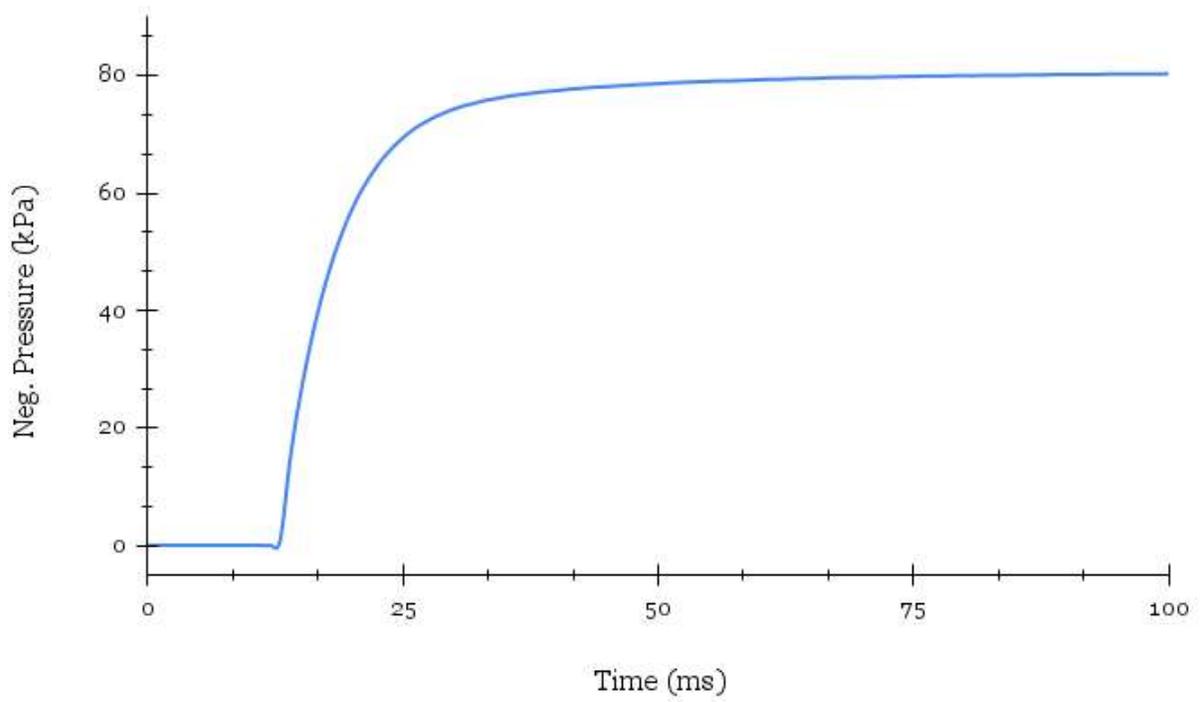


Fig. 44. Pressure against time plot to find rise time for output Oo of the 4-bit pneumatic demultiplexer.

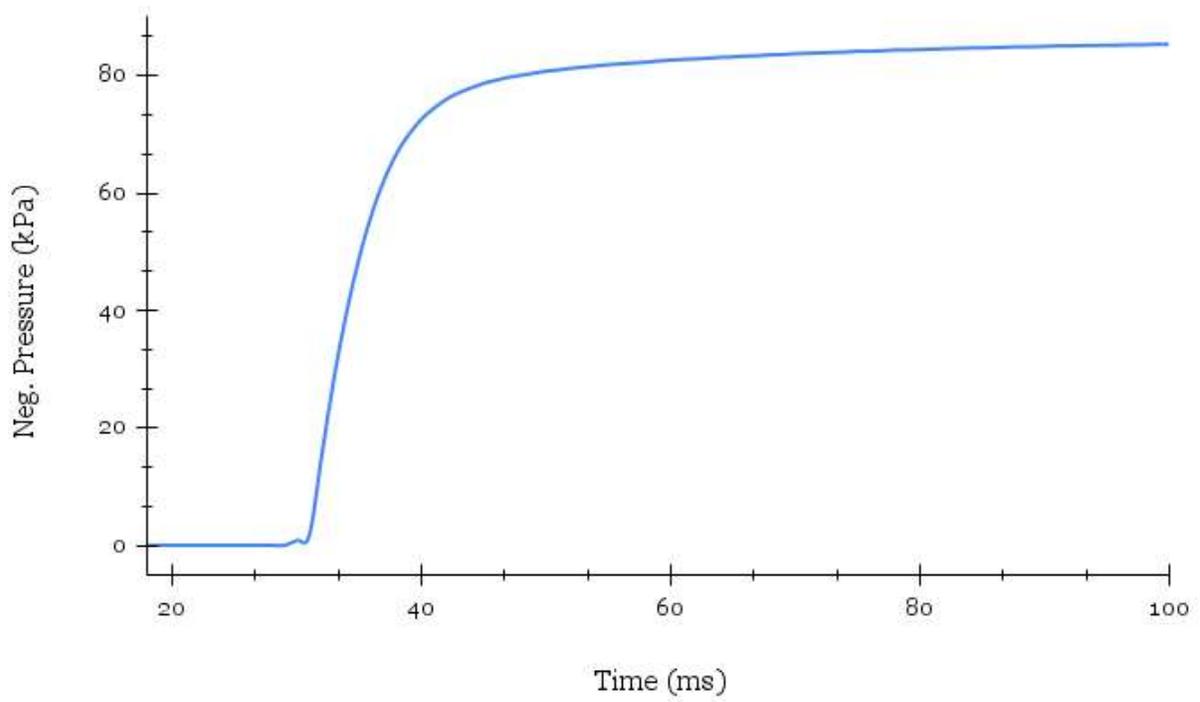


Fig. 45. Pressure against time plot to find rise time for output O6 of the 4-bit pneumatic demultiplexer.

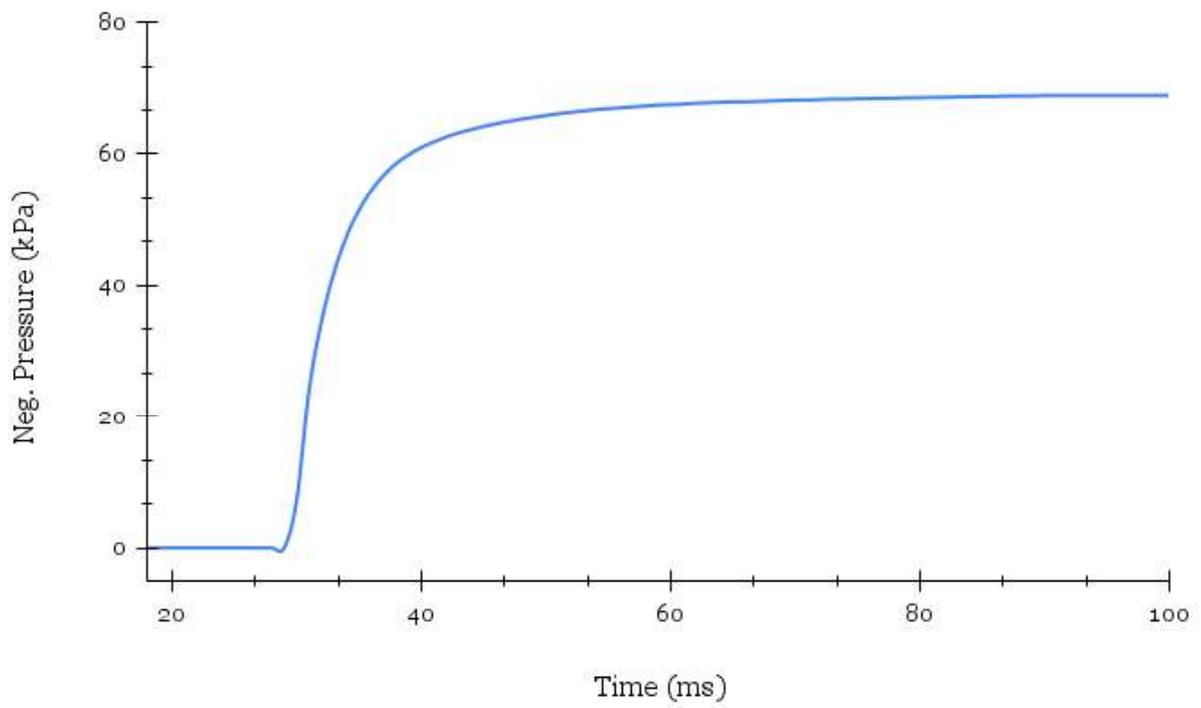


Fig. 46. Pressure against time plot to find rise time for output O10 of the 4-bit pneumatic demultiplexer.

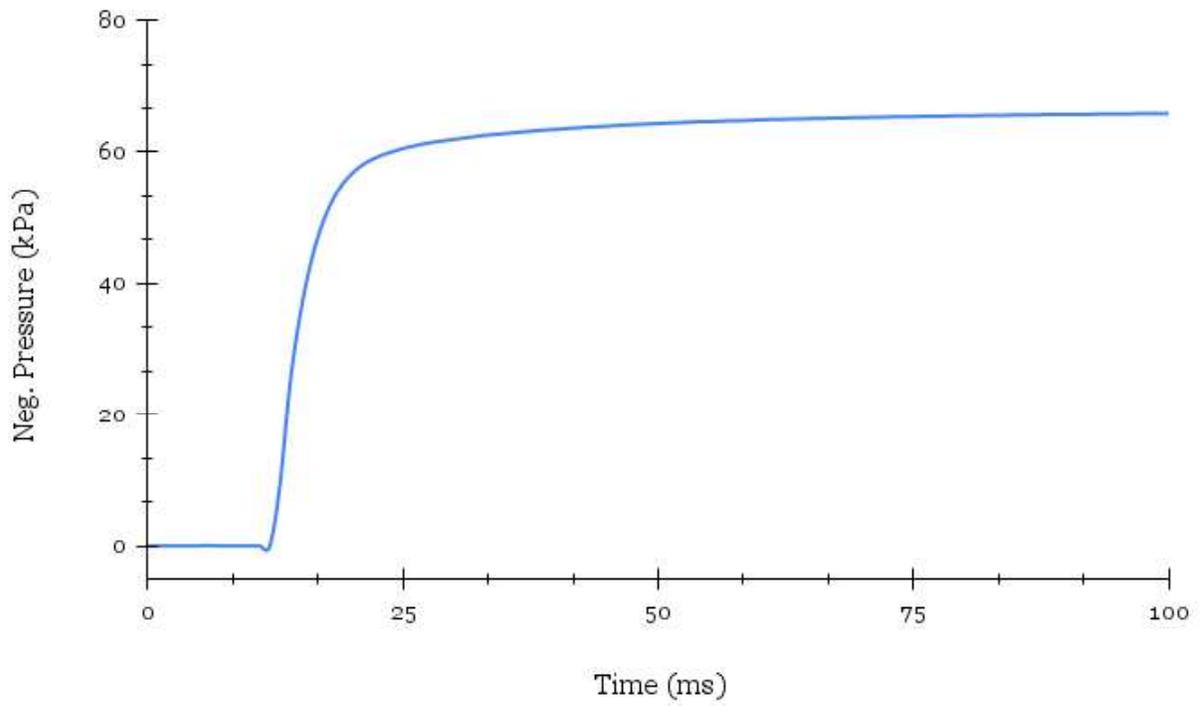


Fig. 47. Pressure against time plot to find rise time for output O14 of the 4-bit pneumatic demultiplexer.