[[1]](#footnote-1)

Logic Gates Using MEMS Devices

A. Iseman, J. Nelson, and C. Olson, *University of Washington Electrical Engineering, Professional Master’s Program*

*Abstract*—Logic gates are the basis of the digital technology used every day. As the interest of the students’ in the EE 503 final project group aligns with embedded hardware, software, and digital logic design, it was natural to extend the study of MEMS into the subject of logic gate construction using MEMS structures and technologies. A half-adder was created in the implementation of this project as a demonstration of MEMS and logic device implementation. This paper encapsulates the development of the subject, implementation, and results of the EE 503 final project.

*Index Terms*—AND logic gate, Coventor, Half-Adder, Microelectromechanical (MEMS) Systems

# INTRODUCTION

F

UNDAMENTAL digital logic gates are the basis of modern technology. Students who undertake the study of digital technology and design are introduced to logic gates early in their educational career as an undergraduate. These gates are AND, OR, NAND, NOR, XOR, NOT, and XNOR gates. Single logic gates can be joined to make larger designs and structures; these structures subsequently are used in numerous locations throughout processors and computers. Normally constructed from transistors and electrical circuit parts, exploring the implementation of digital circuits using microelectromechanical systems (MEMS) was an interesting subject to explore for a final EE 503 project. Additionally, it was natural for students with interests relating to embedded hardware and software to explore the synthesis of mechanical systems and digital systems.

The cantilever tutorial from EE 503 was the basic unit for this project. Properties of this small mechanical system were manipulated such that the cantilever beam would either make contact or not make contact with the output voltage electrode. This contact would then cause the system to understand a high voltage output (a logic 1) or a low voltage output (a logic 0). This can be used to create units that act as logic gates. Manipulation of input voltages on electrodes of the cantilever system can be manipulated to form various gates. Logic gates can form infinite variations of systems and logic equations.

Processors and large digital systems are comprised of many different systems that use these logic gates. Mathematical units such as the arithmetic logical unit (ALU) are examples of larger applications of logic gates. The most basic mathematical process that these units execute is the addition of two numbers. However, mathematical units can execute many other functions as well. Another mathematical process that ALU and mathematical units can execute is subtraction. Addition is often achieved through adder units.

Adders perform the addition of two numbers, as one may ascertain from the name of the unit. An adder is a fairly simple unit that can be constructed numerous ways. It is commonly created from two XOR gates, two AND gates, and an OR gate. Using small, individual mechanical parts to construct logic gates, a full adder was difficult to implement given limitations of software and the time given for the project at the end of the term. A half-adder was less intensive to design and implement and could be built upon in order to implement a full adder. Two half-adders can be used to complete a full-adder structure and the half-adder only uses one XOR gate and one AND gate. However, this can also be achieved with five NAND gates and one NOT gate.

In this project, digital logic gates and a half-adder were explored and implemented. This project report will outline how the NOT gate, NAND gate, AND gate, and half-adder were implemented in Coventor. Additionally, it will also discuss the difficulties of implementing the design and the results of implementation. Finally, this report will discuss lessons learned through the course of the project.

# Implementation

## NOT Gate

The NOT gate was the most simple logic gate to implement in the process of this project. Because it only inverts an input, as shown in Section IIIA, only one output and one input is needed. Thus, there is one input and one output electrode. This is shown in Figure 1.

To achieve the desired output of a NOT gate with the cantilever system, a constant voltage () is applied to the cantilever beam. Either 0 V or V is applied to the input electrode. When the voltage on the beam matches that of the beam, it will deflect and not make contact with the output electrode. In this case, the system outputs a 0 V output and this is interpreted as a logical low. When the system has applied to the beam and 0 V applied to the input electrode, the beam bends and makes contact with the output electrode. The system outputs a output and thus a logical high.

This was consistent with the logic table in Section III. When you apply a logical high to a NOT gate, the result is a logical low output. Conversely, when a logical low is input into a NOT gate, a logical high is output. A NOT gate can also be seen Figures 7, 8, and 9.

## NAND Gate

The NAND gate was more complicated to implement than the NOT gate. A NAND gate functions like an AND gate; however, the output of the NAND gate is the negated AND gate output. This can be achieved through an AND gate and NOT gate in series.

Unlike the NOT gate which negates one input, the NAND negates the output of two inputs. Thus, one electrode is not appropriate in the implementation of the NAND gate. It is required that there are two electrodes instead of one to apply different voltages to in order to attain varying outputs. A NAND structure can be seen in Figures 2, 5, and 6. The varying input voltage each provides a different result. The table showing the varying inputs and outputs is shown in Section IIIB.

Simple cases to visualize the mechanical displacement of the cantilever beam are when the cantilever beam has an applied voltage equal to that of the input electrodes and when the cantilever beam has a voltage applied and no voltage is applied to the two input electrodes. When is applied to the beam and is applied to electrode A and electrode B (our two input electrodes), the beam displacement is away from the output electrode. When is applied to the beam and 0 V is applied to both of the electrodes, the beam displacement is toward the output electrode. The displacement of the cantilever beam away from the output electrode is a logical low, or a logical zero; the displacement of the cantilever beam toward the output electrode is a logical high, or a logical one.

This behavior can be seen in Figures 5 and 6. In these simulations from Coventor, it is clear that the most displacement is concentrated at the end of the beam where the structure has an upward or lower displacement, depending on the applied input voltages.

## AND Gate

Once the NAND gate was implemented, the AND gate is a simple combination of the NOT and the NAND gates. With these two gates in sequence, the NAND gate’s output is negated such that the output of the two gates together is the output of an AND gate. The output of these are in conjunction with the output of an AND gate by itself. This table of inputs and outputs can be seen in Section IIIC.

The structure of the AND gate can be seen in Figure 3. In this layout, the output of the NAND gate () is connected to the input () of the NOT gate, and the output of the NOT gate () is the output of the AND gate.

## Half-Adder Implementation

A half-adder is a combination of digital logic gates that when combined, perform the addition of two numbers. Usually, these are designed with one XOR and one AND gate. Half-adders take two inputs, A and B, into the system and produce two outputs, a sum and a carry bit. In this implementation, five NAND gates and one NOT gate are used to produce the same result. Four of these NAND gates produce the sum bit and the other NAND gate in series with the NOT gate produces the carry bit of the half-adder.

The inputs A and B are connected to the various NAND gate inputs as shown in Figure 4 and the wiring and connections between electrodes of the NAND gates is shown in Figure 9. When traced, these wire routes match that of the diagram in Figure 4. Four of the five NAND gates implemented in Coventor are used to produce the sum output and one is used in series with a NOT gate to produce the carry output. The truth table used to develop this structure can be seen in Section IIID.

While simulating a single unit was fairly simple, simulating a cascading system of units was exponentially more difficult in Coventor. Thus, with the complexity of the project and the time allotment of the project, a simulation of the full half-adder system was not feasible. However, using the singular simulated unit and drawing the system out on paper, the group had confidence that if simulated, the system would produce the desired digital logic output and the correct mechanical displacement in order to achieve this.

## Figures

Below are figures generated in the development of this project and the development of the presentation for the final project of this class.

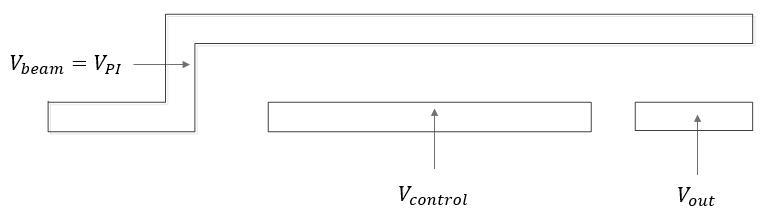


Fig. 1. A Microsoft Visio drawing of the cantilever structure for a NOT gate. Note that there is only one electrode for the input and one electrode for the output.

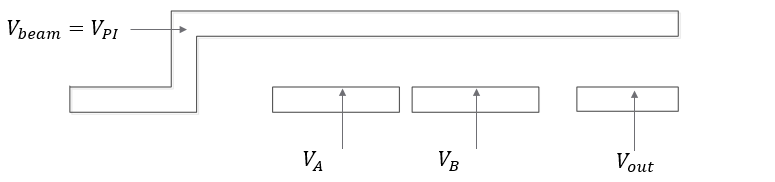


Fig. 2. A Microsoft Visio drawing of the cantilever structure for a NAND gate. Note that there are two electrodes for the inputs and one output electrode.

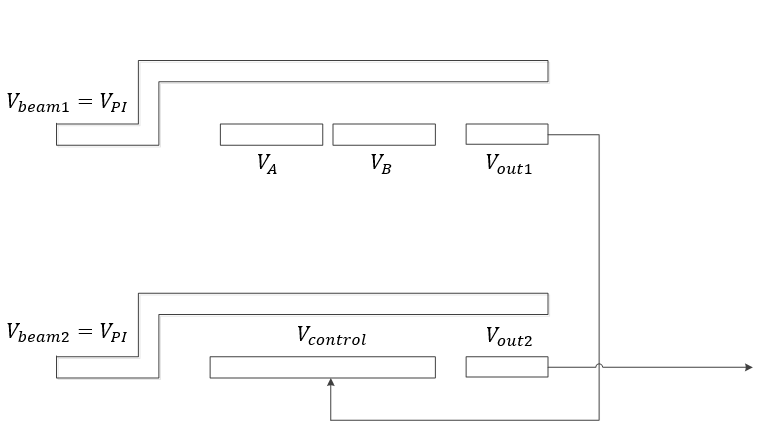
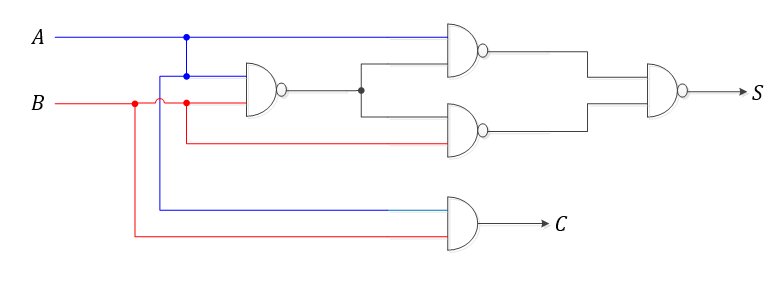


Fig. 3. A Microsoft Visio drawing of the cantilever structure for a AND gate. The NAND and NOT gates are in series such that the output of the NAND gate is negated. Note that there are two electrodes for the inputs and one output electrode.

Fig. 4. A visual diagram of how a half-adder is drawn with logic gate symbols.

## Electronic Image Files

Below are images from the design and implementation of the half-adder in this project.

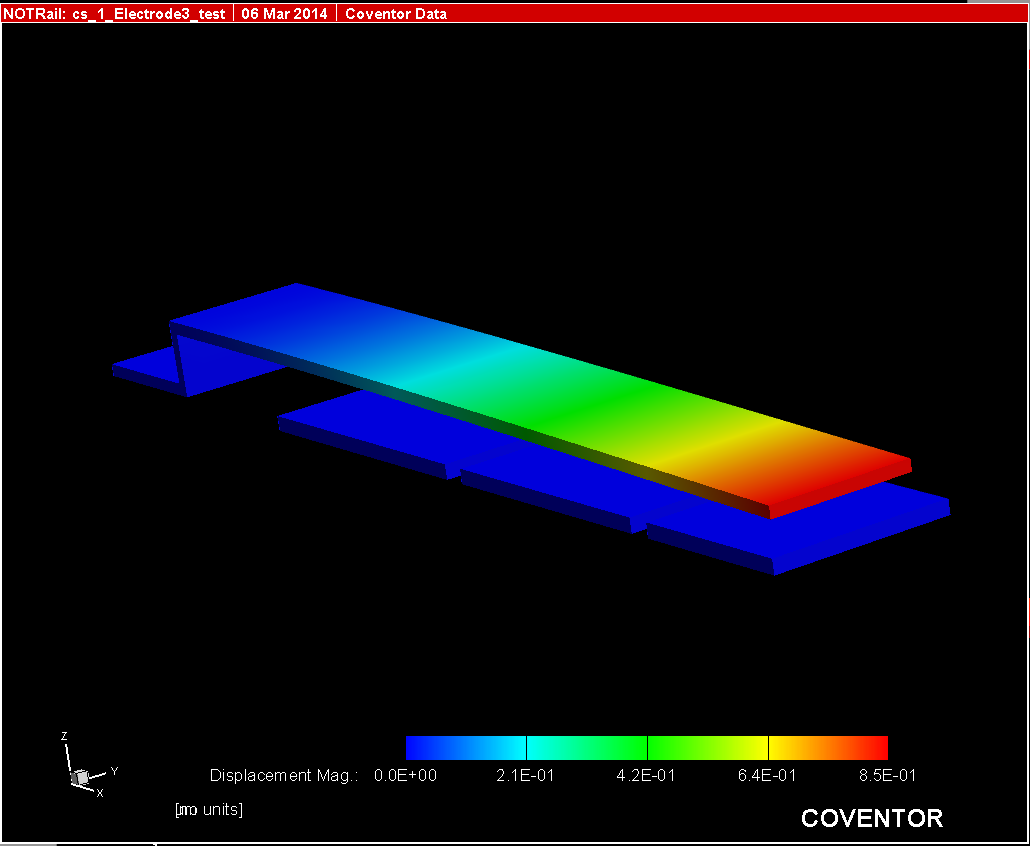


Fig. 5. A singular mechanical cantilever unit is shown in Figure 1. In this instance, the mechanical device is acting as a NAND gate. 60 V is applied to the beam and both electrodes in this simulation and the beam does not touch the output voltage electrode. This signifies a logic “low,” or a “0.”

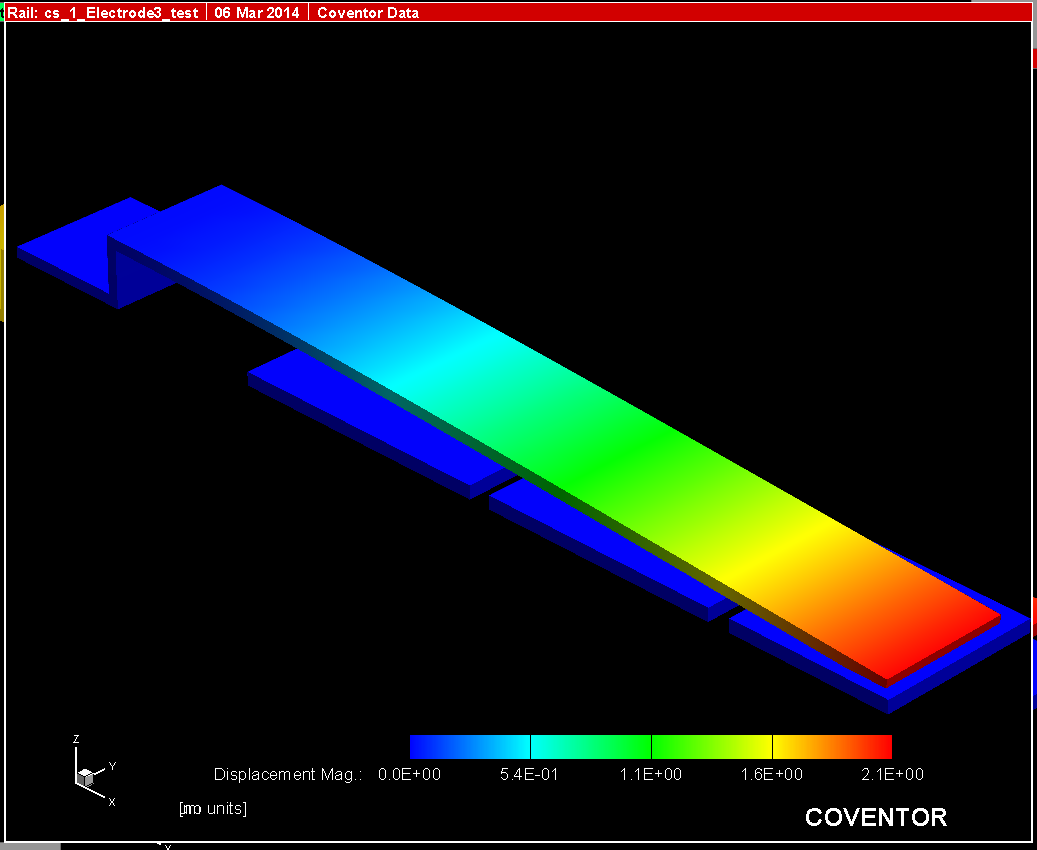


Fig. 6. A singular mechanical cantilever unit is shown in Figure 2. In this instance, the mechanical device is acting as a NAND gate. 60 V is applied to the beam and 0 V is applied to both of the input electrodes. The beam makes contact with the output voltage electrode, signifying a logic “high,” or a “1.”

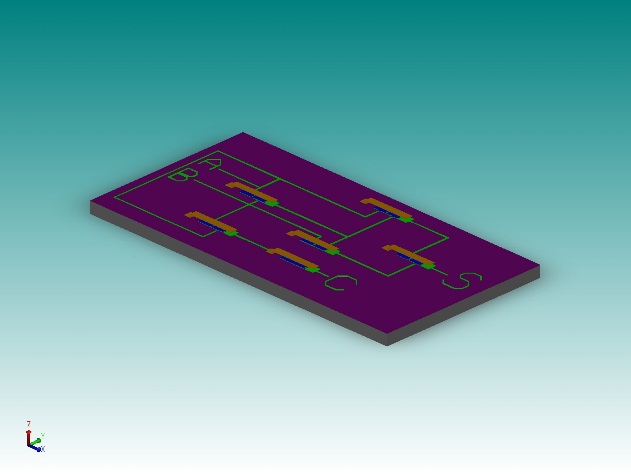


Fig. 7. The complete implementation of the half-adder. As shown, five NAND gates and one NOT gate are required to implement a half-adder. This is the view of the half-adder showing the full implementation of the design with inputs and outputs labeled, as well as the wiring shown.

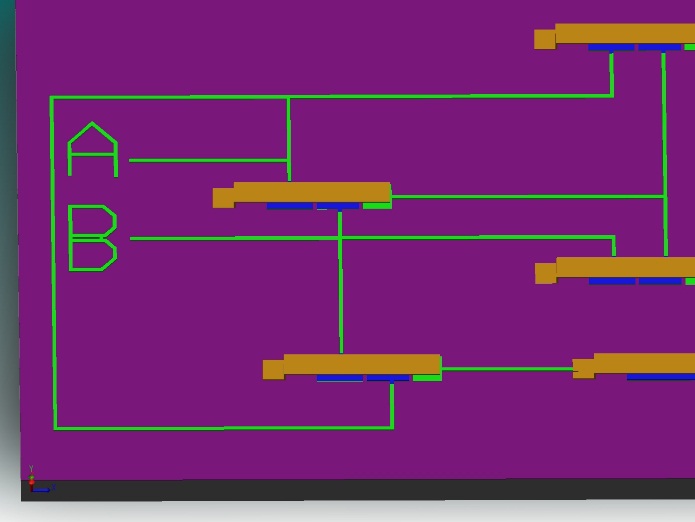


Fig. 8. A more detailed view of the half-adder inputs is shown in Figure 4. The NAND gates and wiring connecting the system together are clearly visible. Also, the difference between the NAND and NOT input electrodes can be seen on the left (NAND gates) and far lower right (NOT gate).

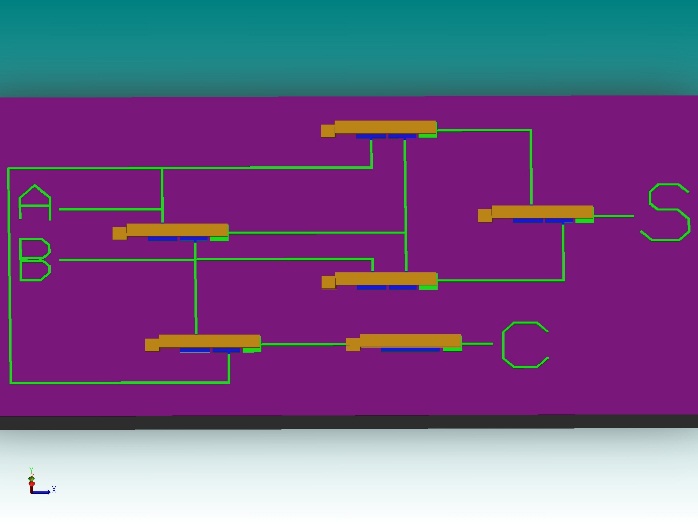


Fig. 9. A more detailed view of the full implementation of the half-adder is shown in Figure 5. The NAND gates, the NOT gate, inputs, and outputs are shown clearly in this image capture of the half-adder system from Coventor.

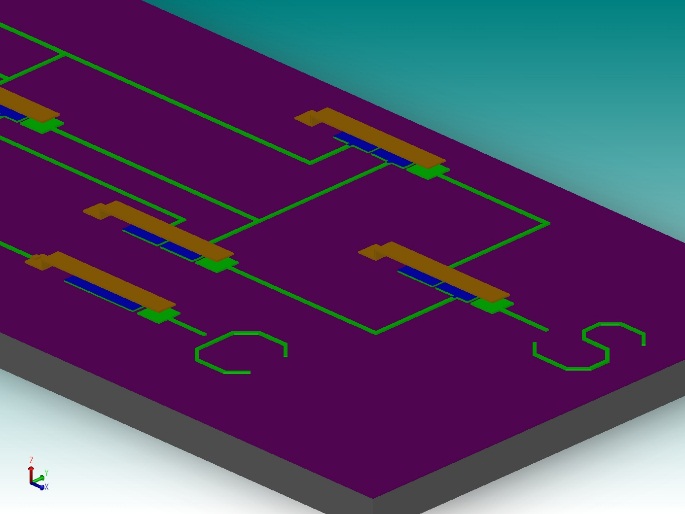


Fig. 10. A more detailed view of the outputs of the half-adder, the sum and carry bits.

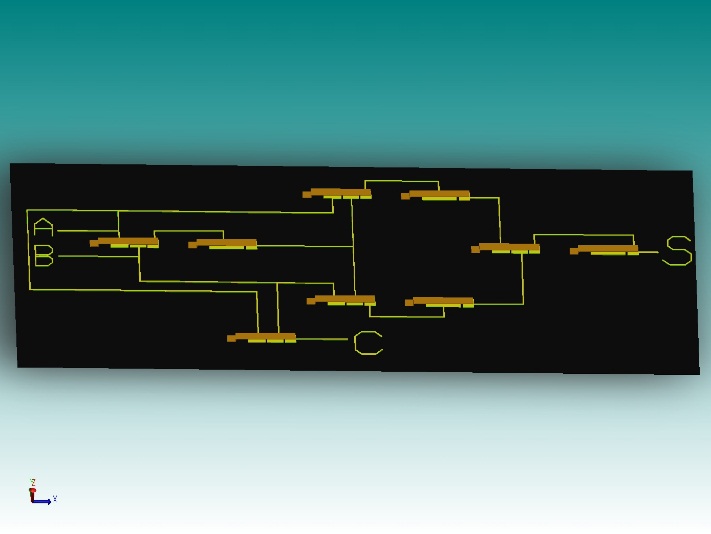


Fig. 11. The full implementation of the half-adder as originally designed. It was thought that originally AND gates were implemented, which required NOT gates following the AND to implement a NAND gate. However, this proved not to be true and the NOT gates were removed following all NAND gates except for that of the carry bit.

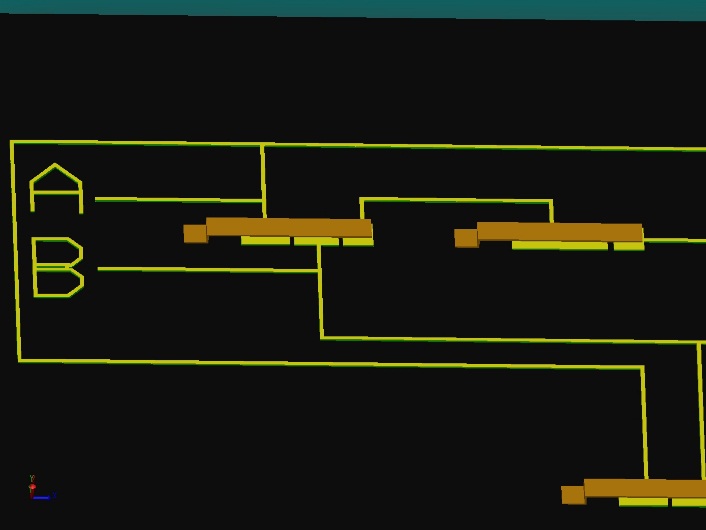


Fig. 12. A detailed view of the original implementation of the half-adder. The first gate on the left part of Figure 8 was thought to be an AND gate. Thus, a NOT gate was required to execute a NAND gate. However, this proved not to be the case and the following NOT gate was removed from following the NAND gate to result in the correct logical outputs.

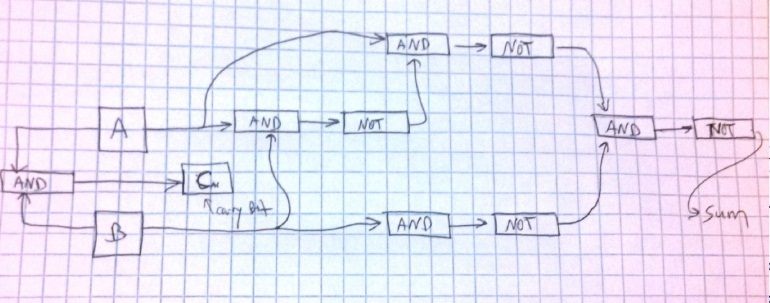


Fig. 13. The original design for the half-adder as drawn before implementation. Instead of implementing an AND gate and then requiring a NOT gate after this gate, a NAND was easier to implement. The only gate that then required the NOT gate following the NAND was the two gates (the AND gate) that produces the carry bit.

# MATH

The logic tables that were used to develop and verify the design of this half-adder system have been included below per type of gate that was implemented and per the final half-adder implementation.

## NOT Gate

|  |  |
| --- | --- |
| Input | Output |
| 0 | 1 |
| 1 | 0 |

## NAND Gate

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## AND Gate

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Half-Adder

|  |  |  |  |
| --- | --- | --- | --- |
| Input A | Input B | Carry | Sum |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Results

The first unit to be implemented for this design was the NOT gate. This was the simplest gate to implement due to the fact that a NOT gate only inverts a signal; only one electrode is needed. For other gates, there is often two inputs that are compared and an output is interpreted accordingly. These gates need two input electrodes in order to produce different outputs based on different combinations of inputs. The NAND gate was one such gate. There was significant guessing and running the Coventor simulation of the cantilever beam with two electrodes to determine the size of input electrode that would cause the desired displacement of the beam toward or away from the output voltage electrode.

A single unit was straightforward to implement and simulate. These individual units have a small number of meshes and layers that need to be named in the Coventor simulation in order to allow it to build and compile. However, it was quickly realized that as individual units were added to the design and the system, the number of meshes and layers that needed to be named in order for the simulation to run correctly. In addition to having this requirement for the simulation to run, the increased number of layers and meshes also causes the simulation to take longer to process and execute. It was because of this that when the group implemented the half-adder, it was decided that it would be too laborious and take too much time to try to make the simulation work. By looking at the simulation of each unit, the group could verify whether or not each unit would behave as expected.

The original design that the group had drawn out for the half-adder is shown in Figure 12. The NAND gate was much easier than the AND gate to implement; however, the group did not realize that the NAND had been fully implemented versus the AND gate, so the outputs of the gates were being inverted and did not appear correct at first implementation. Once it was realized that the NOT gates could be removed from being in series with all of the NAND gates, the half-adder could produce the correct and expected output. (Again, this was not simulated, but each unit was observed based on a single unit implementation and simulation to have the expected output to get the correct half-adder output.) The only NOT gate that remained in the design was that in series with a NAND gate to produce the carry bit output. These original implementations can be seen in Figures 11, 12, and 13.

Overall, this project was fairly successful and interesting to explore during the remainder of EE 503. None of the members of the group had used a mechanical system in place of traditional designs before, so it was a challenge to initially figure out and then once understood, was simple to implement. While software, especially Coventor, can simulate a large number and types of programs, there are still limits to how fast or easily a simulation could be run. If there were more time to explore this project, the group would have liked to explore the construction of an OR gate and further develop a simulation of the half-adder system. All individual units and logic gates were successful and operated as expected.

# Conclusion

In this final project for EE 503, the students in this group diverted from the traditional design of logic gates with transistors and developed a mechanical logic gate with a metal beam that moved depending on the charge that it was exposed to on its input electrodes. The first logic gate, and the most simple logic gate, implemented was the NOT gate, followed by the NAND gate, and the AND gate. These were used to complete a half-adder design. Each of the gates were simulated and these simulations were used to determine whether or not the half-adder would indeed work. This was due to the simulation being prohibitively time consuming and complex for this type of a system design. Students in this group learned more about MEMS systems and how to apply them as well as the limitations of software when used for complex design problems. Overall, a great deal was learned about the synthesis of MEMS devices and digital design, so the group considers this project to be a success.

References

1. H. H. Porter. (2007). *The Design of a Relay Based Computer.* Online. Available: <http://www.allaboutcircuits.com/vol_4/chpt_9/2.html>
2. *A Half-Adder*. Online. Available: <http://www.allaboutcircuits.com/vol_4/chpt_9/2.html>

1. A. Iseman is with The Boeing Company, Seattle, WA. He is studying in the University of Washington’s Electrical Engineering Department to attain his M.S.E.E. (e-mail: imadami@gmail.com)

   J. Nelson is with Nytec, Redmond, WA. He is studying in the University of Washington’s Electrical Engineering Department to attain his M.S.E.E. (e-mail: jonah.nelson@hotmail.com)

   C. Olson is with The Boeing Company, Kent, WA. She is also studying in the University of Washington’s Electrical Engineering Department to attain her M.S.E.E. (e-mail: ceolson@uw.edu) [↑](#footnote-ref-1)