Alexander Buchholz

CPE 690

Project

**Pseudo-Random Noise Generator**

Spring 2017

Contents

[Table of Figures 3](#_Toc482054060)

[1. Motivation 4](#_Toc482054061)

[2. Hierarchy Breakdown and Design Considerations 4](#_Toc482054062)

[Overview 4](#_Toc482054063)

[Shift Register Implementation 4](#_Toc482054064)

[XOR2 Implementation 5](#_Toc482054065)

[OR2 Implementation 6](#_Toc482054066)

[3. L-Edit Cell Structure and Simulations 6](#_Toc482054067)

[Shift Register Implementation 7](#_Toc482054068)

[XOR2 Implementation 11](#_Toc482054069)

[OR2 Implementation 14](#_Toc482054070)

[4. Completed Model Structure and Simulations 16](#_Toc482054071)

[5. Difficulties and Concluding Remarks 18](#_Toc482054072)

# Table of Figures

[Figure 1- Pseudo-Random Noise Generator Sketch 4](#_Toc482053997)

[Figure 2 - Shift Register Sketch 5](#_Toc482053998)

[Figure 3 - D Flip-Flop Sketch 5](#_Toc482053999)

[Figure 4 - XOR2 Gate Level Sketch 6](#_Toc482054000)

[Figure 5 - NAND2 Sketch 6](#_Toc482054001)

[Figure 6 - OR2 Sketch 7](#_Toc482054002)

[Figure 7 - Cells Used 7](#_Toc482054003)

[Figure 8 - D Flip-Flop Piece Layout 8](#_Toc482054004)

[Figure 9 - D Flip-Flop Piece Simulation 9](#_Toc482054005)

[Figure 10 - Inverter Layout 10](#_Toc482054006)

[Figure 11 - Inverter Simulation 10](#_Toc482054007)

[Figure 12 - D Flip-Flop Layout 11](#_Toc482054008)

[Figure 13 - D Flip-Flop Simulation 11](#_Toc482054009)

[Figure 14 - Shift Register Layout 12](#_Toc482054010)

[Figure 15 - Shift Register Simulation 12](#_Toc482054011)

[Figure 16 - NAND2 Layout 13](#_Toc482054012)

[Figure 17 - NAND2 Simulation 13](#_Toc482054013)

[Figure 18 – XOR2 Layout 14](#_Toc482054014)

[Figure 19 - XOR2 Simulation 14](#_Toc482054015)

[Figure 20 - NOR2 Layout 15](#_Toc482054016)

[Figure 21 - NOR2 Simulation 15](#_Toc482054017)

[Figure 22 - OR2 Layout 16](#_Toc482054018)

[Figure 23 - OR2 Simulation 16](#_Toc482054019)

[Figure 24 - Noise Generator Layout 17](#_Toc482054020)

[Figure 25 - Noise Generator Simulaton 17](#_Toc482054021)

[Figure 26 - 2.2V Inputs Simulation 18](#_Toc482054022)

[Figure 27 - 100°C Temperature Simulation 18](#_Toc482054023)

[Figure 28 - Fast Reset Simulation 19](#_Toc482054024)

# Motivation

For the final project of CPE690, I chose the Pseudo-Random Noise Generator first described in HW2. Signal processing is a topic that has interested me since my undergraduate studies, so I thought that a noise generator would be something fun to make. I also appreciated the fact that I can re-use the assets and lessons taken from HW2 and HW3.

# Hierarchy Breakdown and Design Considerations

## Overview

Figure 1 shows a sketch of the end goal— a Pseudo-Random Noise Generator with “reset” and “clk” inputs and a 4 bit data vector output. The design process involved breaking the device down into a hierarchy, building the lowest level components, and then assembling the noise generator piece by piece. For this particular design, a XOR2 gate, an OR2 gate, and a Shift-Register were needed.

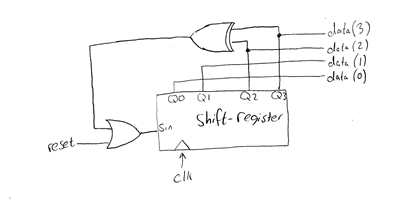


Figure 1- Pseudo-Random Noise Generator Sketch

## Shift Register Implementation

The 4-bit Shift Register shown in Figure 2 is just four D Flip-Flops cascaded together, so the real difficulty of this implementation is designing a D Flip-Flop.

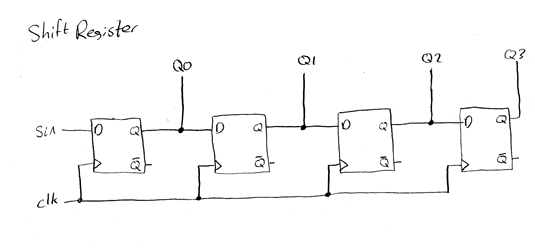


Figure 2 - Shift Register Sketch

Numerous implementations could have been used for the D Flip-Flop, but the design detailed in the Lecture 5 slides was chosen, sketched in Figure 3. This design was easily reducible into pieces, one of which was an inverter, which was needed for the OR2 gate already. With this design, however, an extra inverter was needed to generate an inverted clock, and both clock signals needed to be shared on several nodes across the device.

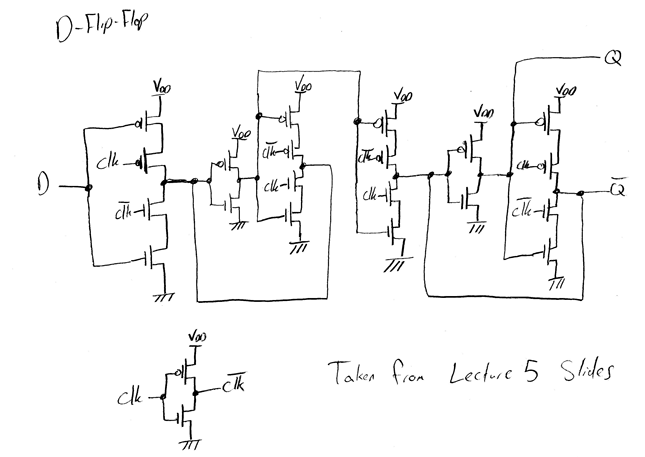


Figure 3 - D Flip-Flop Sketch

## XOR2 Implementation

Like the D Flip-Flop, numerous implementations exist for XOR2 gates. Since Homework 3 involved building a NAND2 gate, the XOR2 implementation chosen consists of four NAND2 gates, sketched in Figure 4. By re-using the NAND2 design from Homework 3, the XOR2 implementation was greatly simplified. For completeness, Figure 5 shows a sketch of the NAND2 transistor layout.

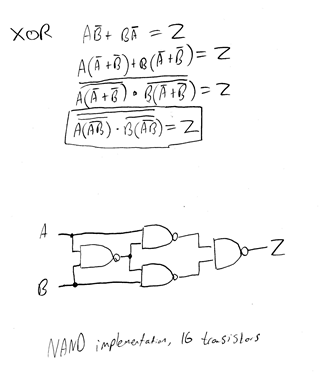


Figure 4 - XOR2 Gate Level Sketch

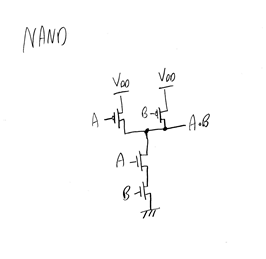


Figure 5 - NAND2 Sketch

OR2 Implementation

The OR2 gate consisted of a NOR2 gate with its output tied to an inverter. Figure 6 shows a simple sketch of the implementation. Because the NOR2 component is the logical inverse of the NAND2 component, the NAND2 L-Edit layout was used as a model in designing the NOR2 layout.

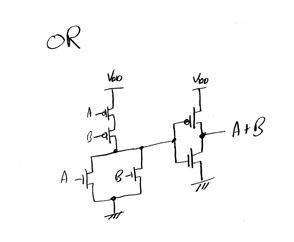


Figure 6 - OR2 Sketch

# L-Edit Cell Structure and Simulations

The sketches in Section 2, Hierarchy Breakdown and Design Considerations, were all laid out in cells and simulated individually in L-Edit. Pre-existing assets from Homework 3 were used when possible. A list of cells is shown in Figure 7, which includes all of the cells used in Homework 3, plus an additional D\_FF\_Piece, D\_FLIPFLOP, INVERTER, M23VIA, NOISE\_GENERATOR, NOR2, OR2, SHIFT\_REGISTER, and XOR2.

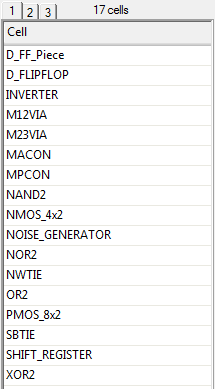


Figure 7 - Cells Used

For all lower-level simulations, a 50fF capacitor was tied to the output, with 3V pulse waveforms added on the inputs. The lower-level simulations were kept intentionally simple, and were used just to verify correct logic on the inputs and outputs of devices.

## Shift Register Implementation

The shift register was designed by cascading multiple D Flip-Flops together. Since the design chosen in Figure 3 was relatively large, with repeatable structures, the D Flip-Flop cell was broken up into two components— a “DFF\_Piece” and an “INVERTER.”

The layout used for the DFF\_Piece is shown in Figure 8, which represents two NMOS and two PMOS transistors in series. Inputs A and B represent arbitrary clock inputs, intended to take either the normal clock or the inverted clock. A simulation of the cell is shown in Figure 9. Although the simulation does show much of the behavior, it shows that the device can drive a strong output in response to inputs.

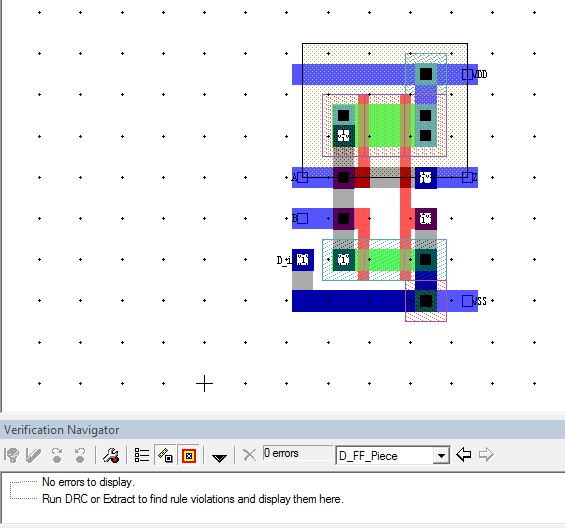


Figure 8 - D Flip-Flop Piece Layout

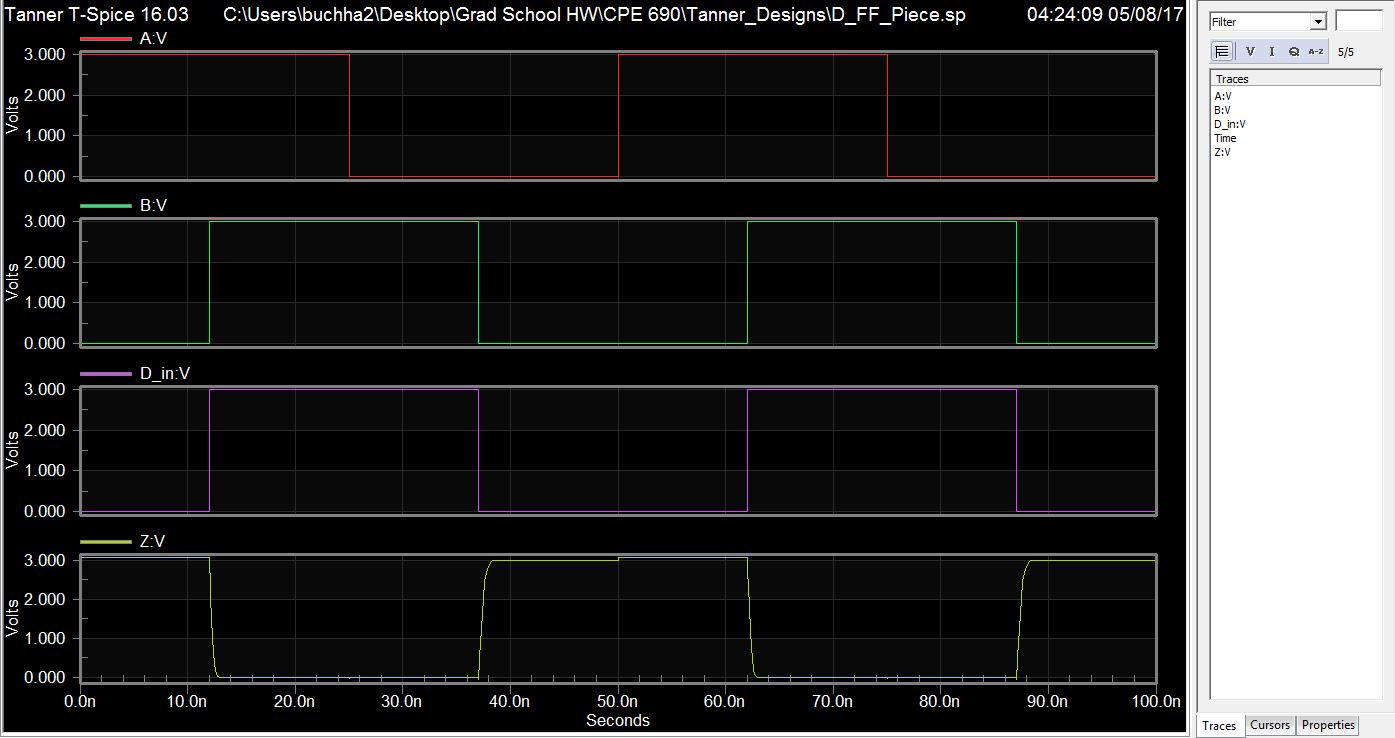


Figure 9 - D Flip-Flop Piece Simulation

An inverter was designed as part of the D Flip-Flop, but was also used in the OR2 gate. For the D Flip-Flop, the inverter was the second “piece” of the larger hierarchy, and was also used to invert the clock signal. A layout is shown in Figure 10, and a simulation in Figure 11. As the simulation shows, the output always shows the expected inverted input.

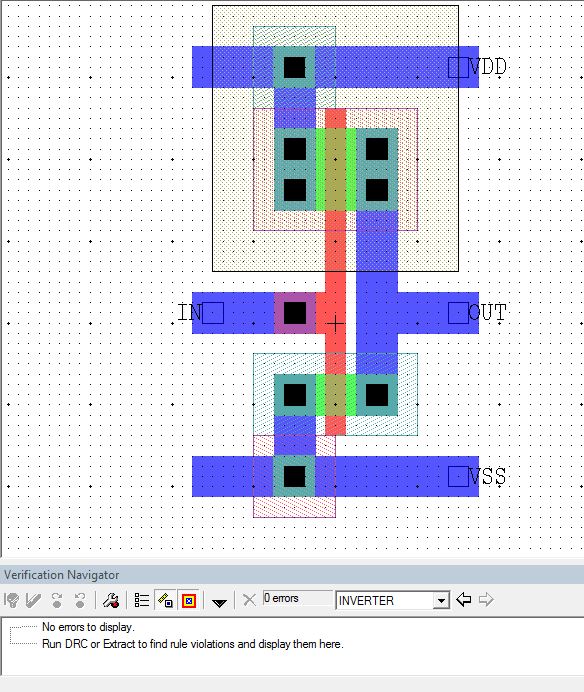


Figure 10 - Inverter Layout



Figure 11 - Inverter Simulation

Several of the D Flip-Flop “pieces” were tied together with the inverters to form the complete D Flip-Flop layout shown in Figure 3. For the clock, an extra inverter was used to generate the inverted clock signal, and both signals were carried along a third metal layer to all of the required terminations; this layout is shown in Figure 12.

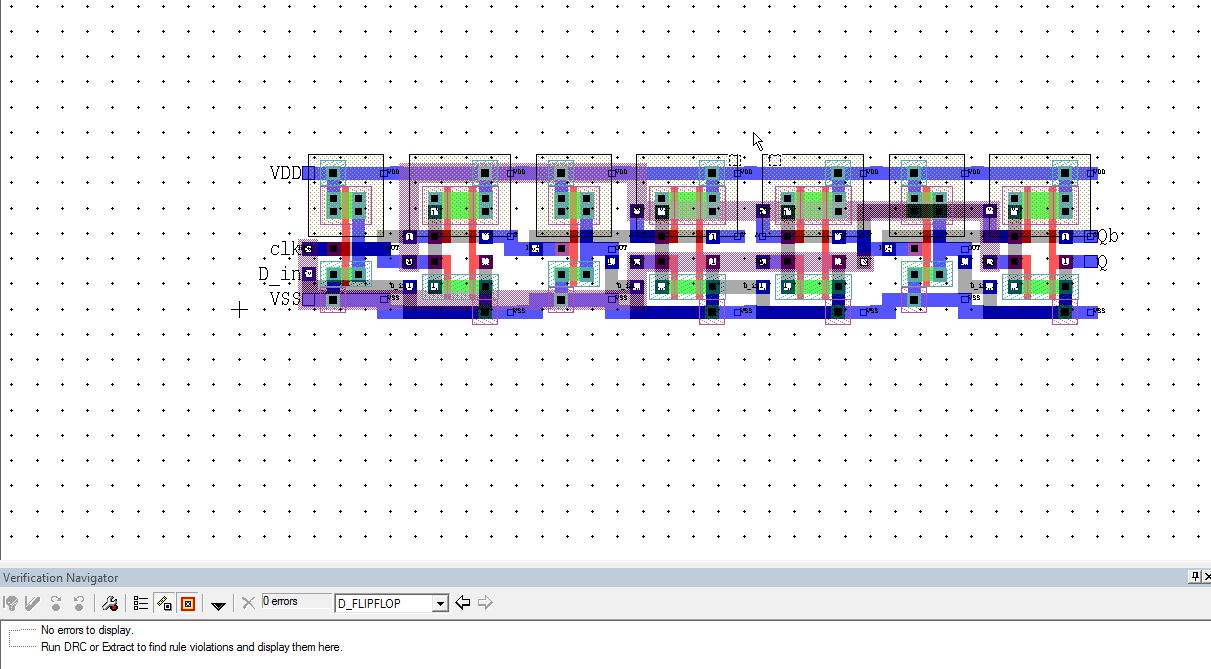


Figure 12 - D Flip-Flop Layout

A simulation of the D Flip-Flop layout is shown in Figure 13. The important takeaway is that the output Q only follows D when the positive edge of the clock is triggered. Additionally, output Qb is an inverted Q, as one would expect.



Figure 13 - D Flip-Flop Simulation

After the D Flip-Flop was generated, constructing the Shift Register was just a matter of cascading four D Flip-Flops together. Because the clock was already carried throughout the D Flip-Flop, there was no need for any additional metal for the “sin” and “clk” signals. The Shift Register layout is detailed in Figure 14, and its simulation in Figure 15. The simulation shows expected behavior, as each output is always one step behind its predecessor.

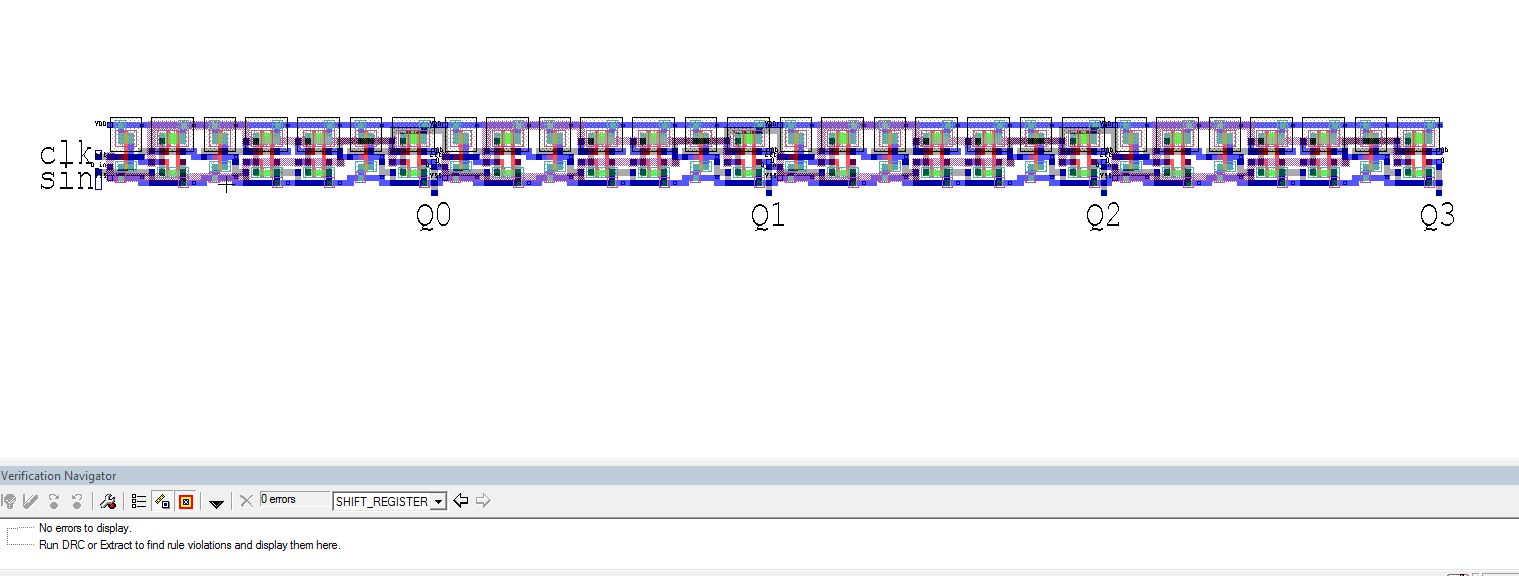


Figure 14 - Shift Register Layout

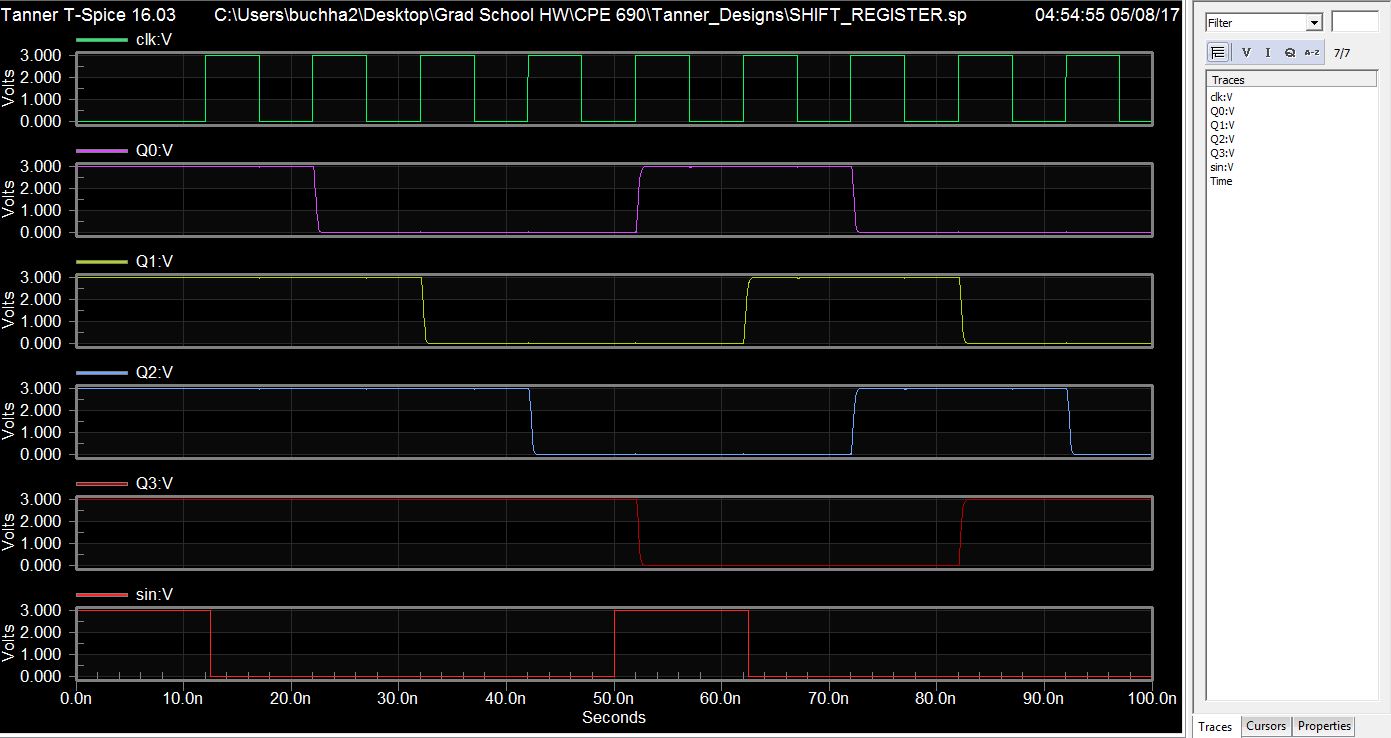


Figure 15 - Shift Register Simulation

## XOR2 Implementation

Because the four-NAND2-implementation was chosen for the XOR2 gate, designing the gate was simple. The NAND2 cell, laid out in Figure 16 and simulated in Figure 17, were taken directly from Homework 3 files, and tied together as detailed in Figure 18. The simulated output in Figure 19 demonstrates the expected logic of a XOR2 gate.

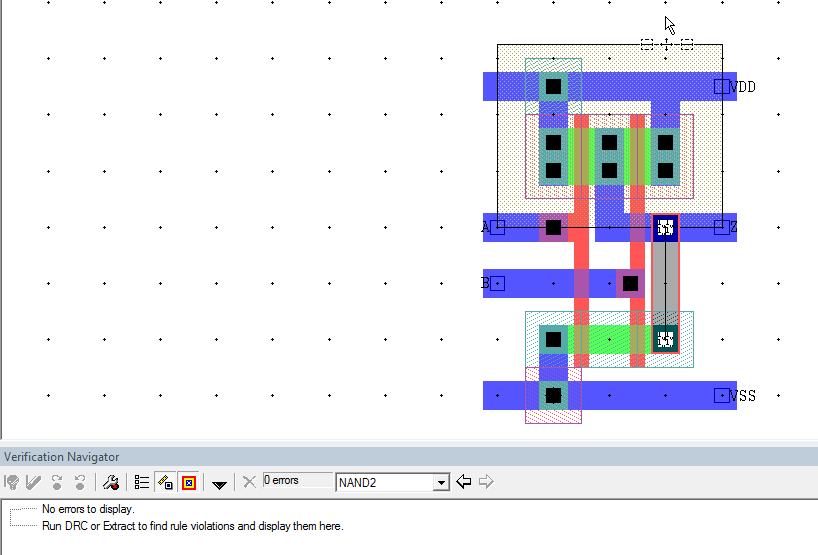


Figure 16 - NAND2 Layout

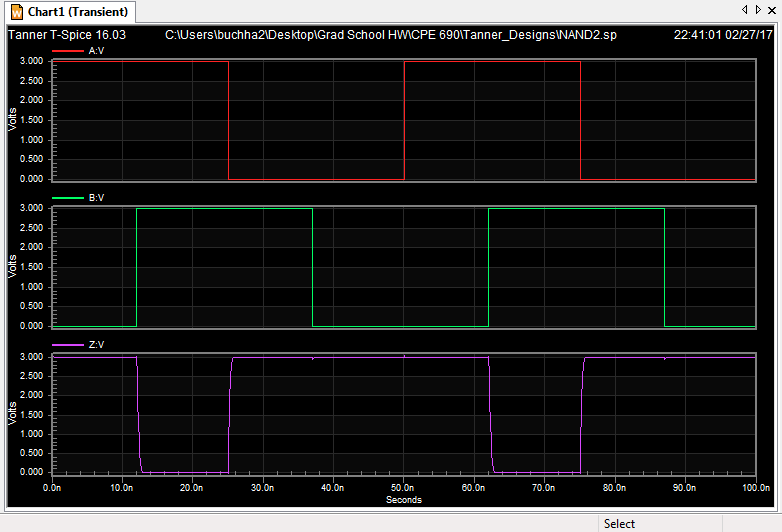


Figure 17 - NAND2 Simulation

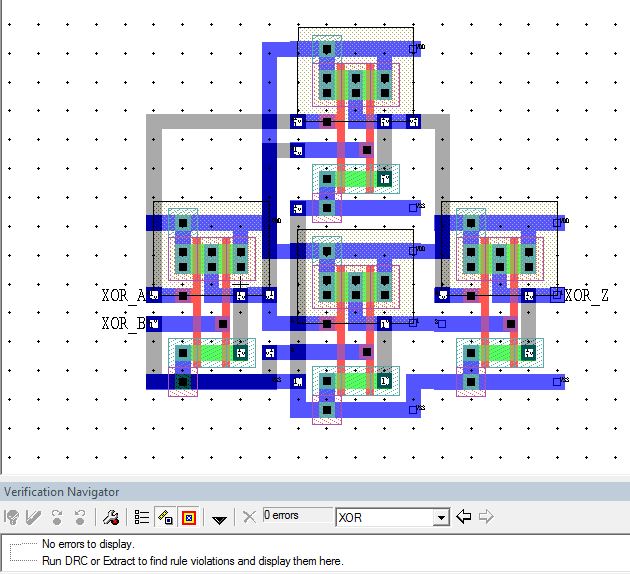


Figure 18 – XOR2 Layout

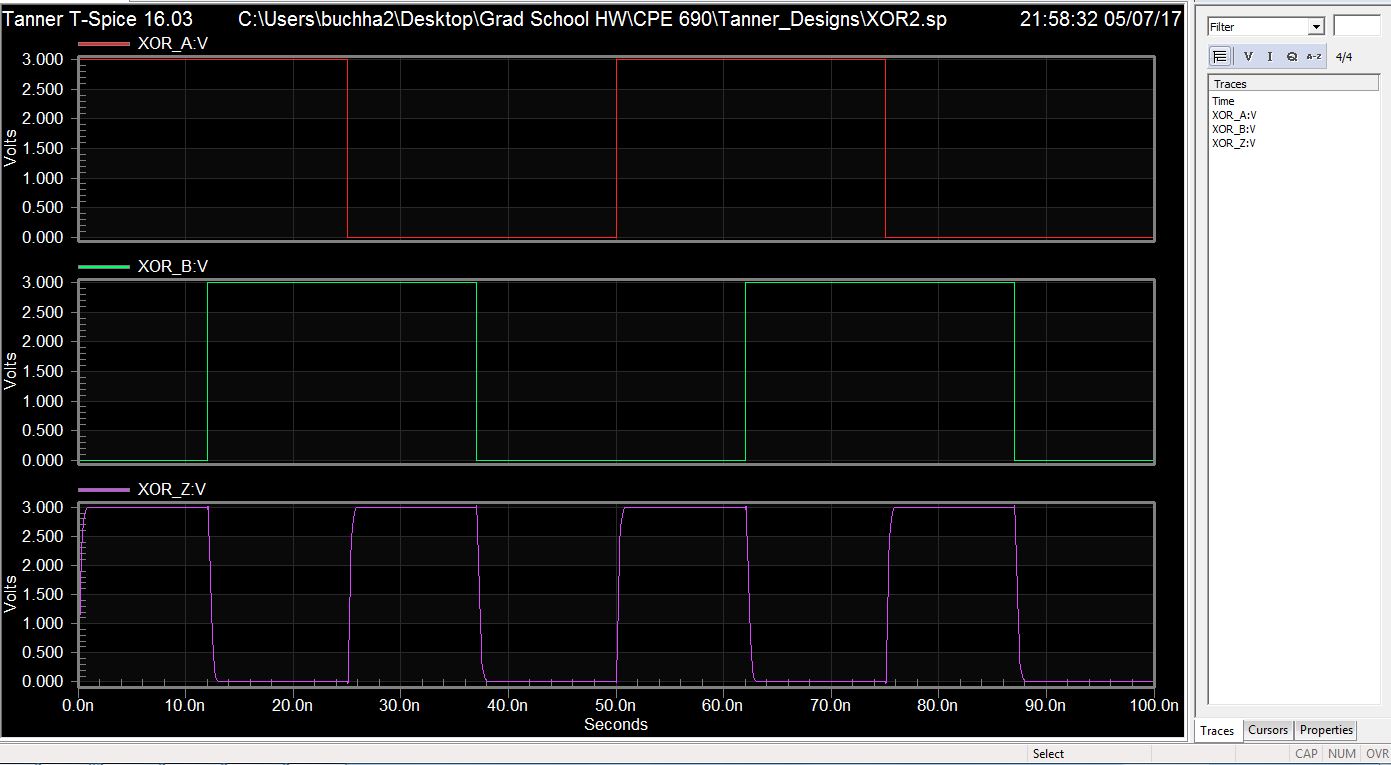


Figure 19 - XOR2 Simulation

## OR2 Implementation

The OR2 gate was constructed by inverting the output of a NOR2 gate. Figure 20 shows the NOR2 layout, which was designed by inverting the logic of the NAND2 layout shown in Figure 16. The simulation in Figure 21 shows the expected logic of a NOR2 gate.

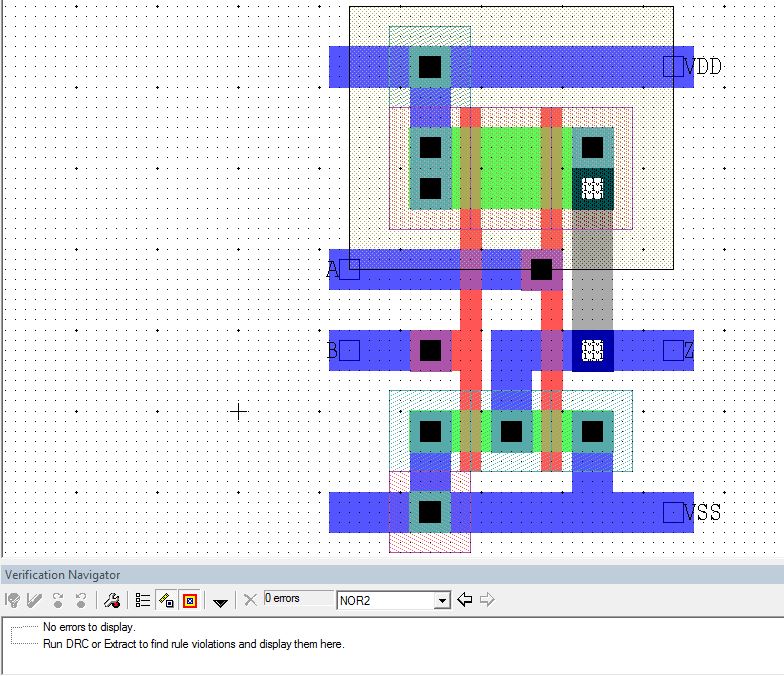


Figure 20 - NOR2 Layout

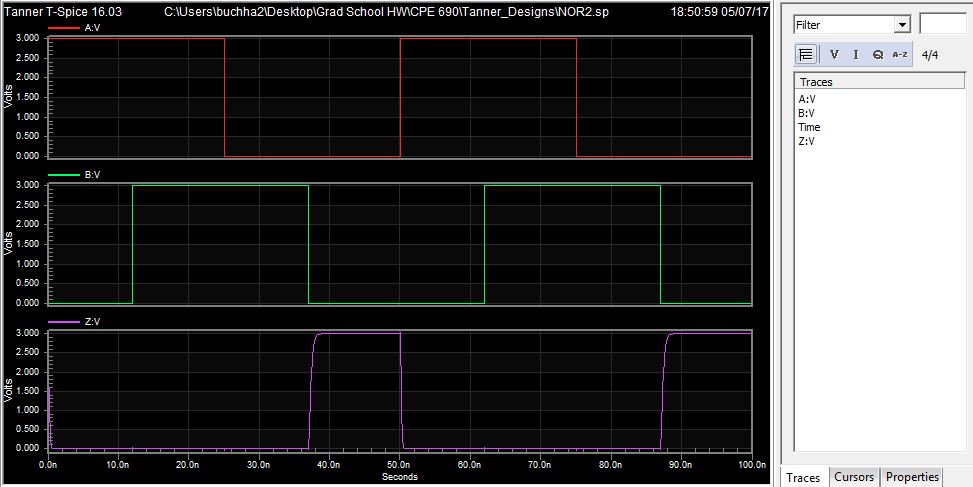


Figure 21 - NOR2 Simulation

The inverter shown in Figure 10 was then attached to the output of the NOR2 to produce the layout and simulation detailed in Figure 22 and Figure 23. The simulation shows the expected logic of an OR2 gate.

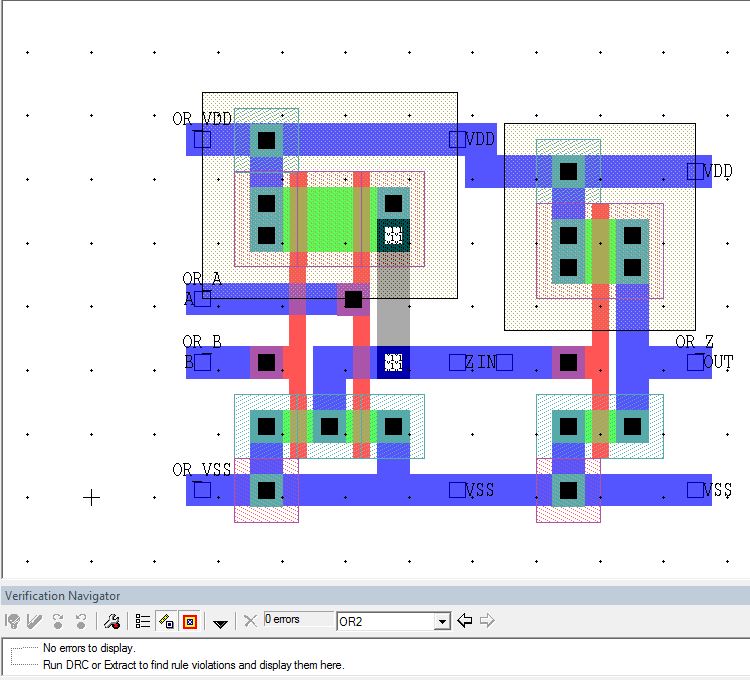


Figure 22 - OR2 Layout



Figure 23 - OR2 Simulation

# Completed Model Structure and Simulations

Finally, the Shift Register, XOR2, and OR2 were hooked together to build the Noise Generator shown in Figure 24. Its output shows a pattern that appears random at first, but it repeatable and predictable, hence “pseudo” random. The pattern also starts over whenever the reset is triggered. Because the noise generator was built from a shift register, the output signal is cascaded and shifted on data 1 through data 4, resulting in the identical noise signal shown on each data line.

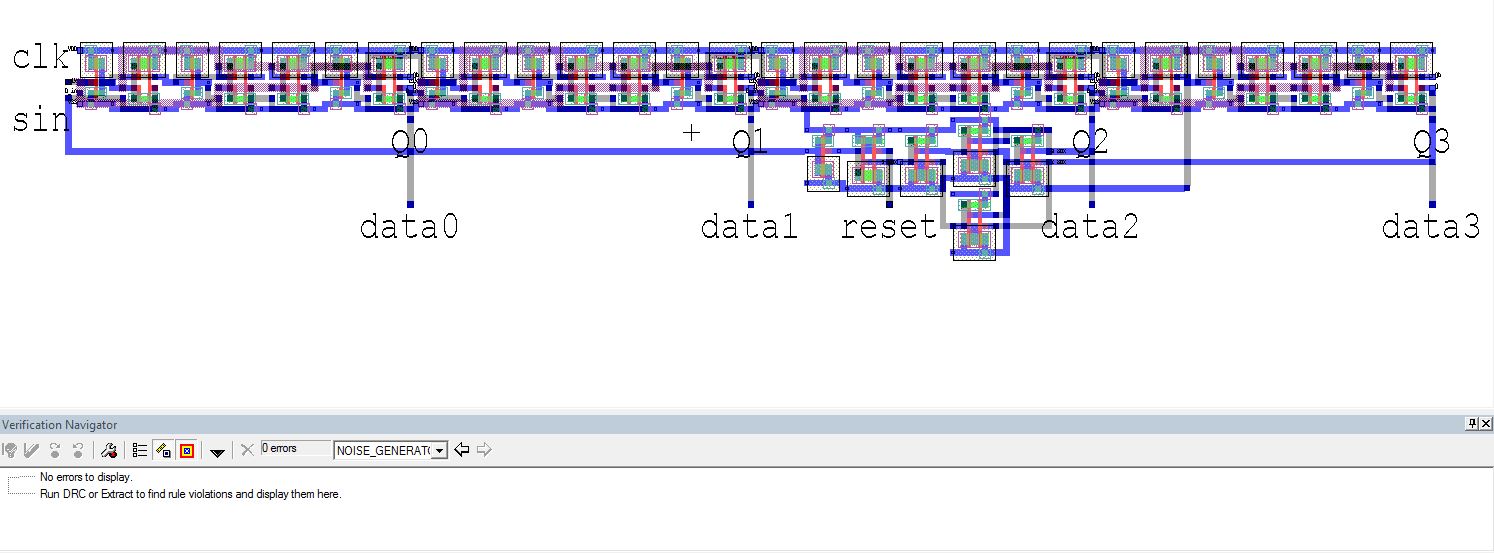


Figure 24 - Noise Generator Layout

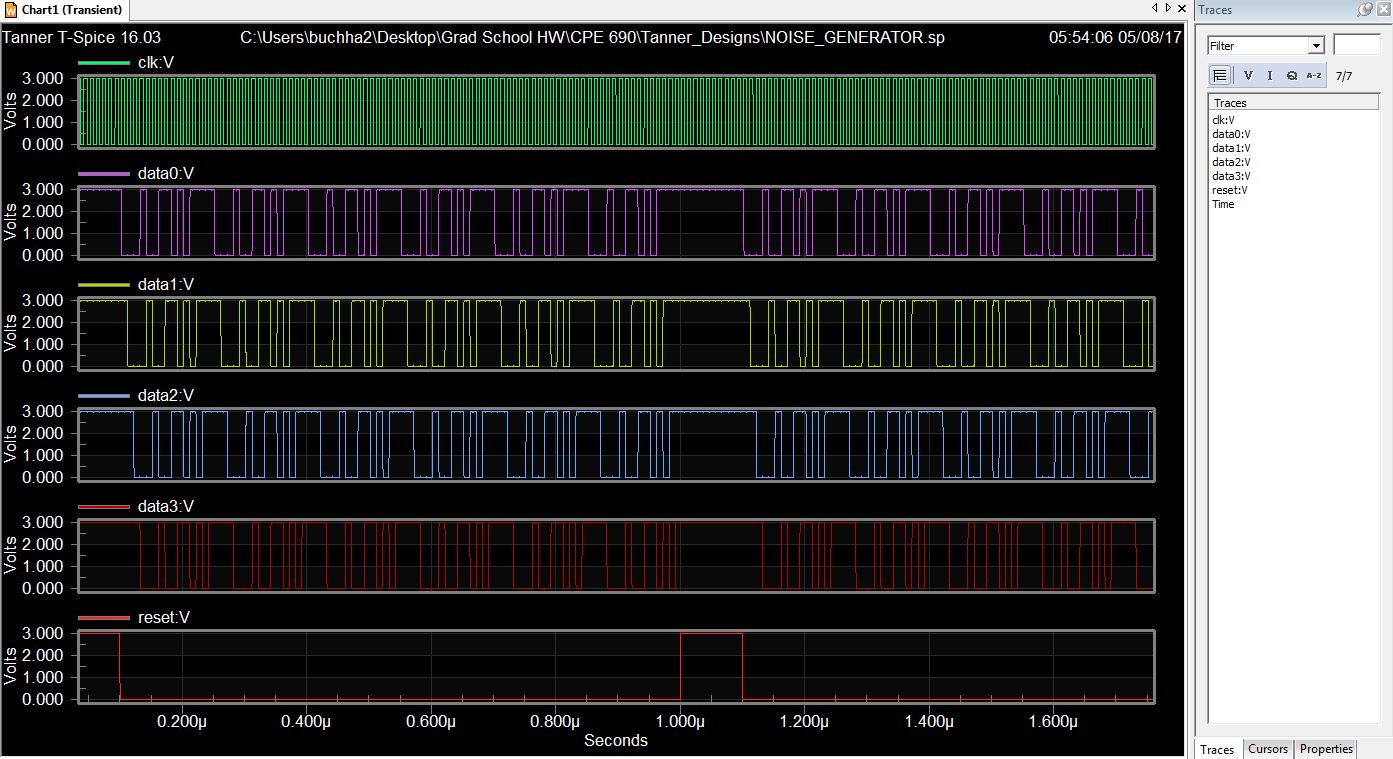


Figure 25 - Noise Generator Simulaton

In addition to the basic simulation, additional tests were run against the completed design. Figure 26, Figure 27, and Figure 28 detail demonstrate scenarios and their various effects. From these few tests, it seems that the circuit is well-behaved under most scenarios, and only extreme scenarios will produce odd results. For example, the fast-reset shown in Figure 28 breaks the “pseudo-random” property. Test scenarios with clock speeds close to the rise-fall time, or any other strange inputs, can also produce strange outputs.

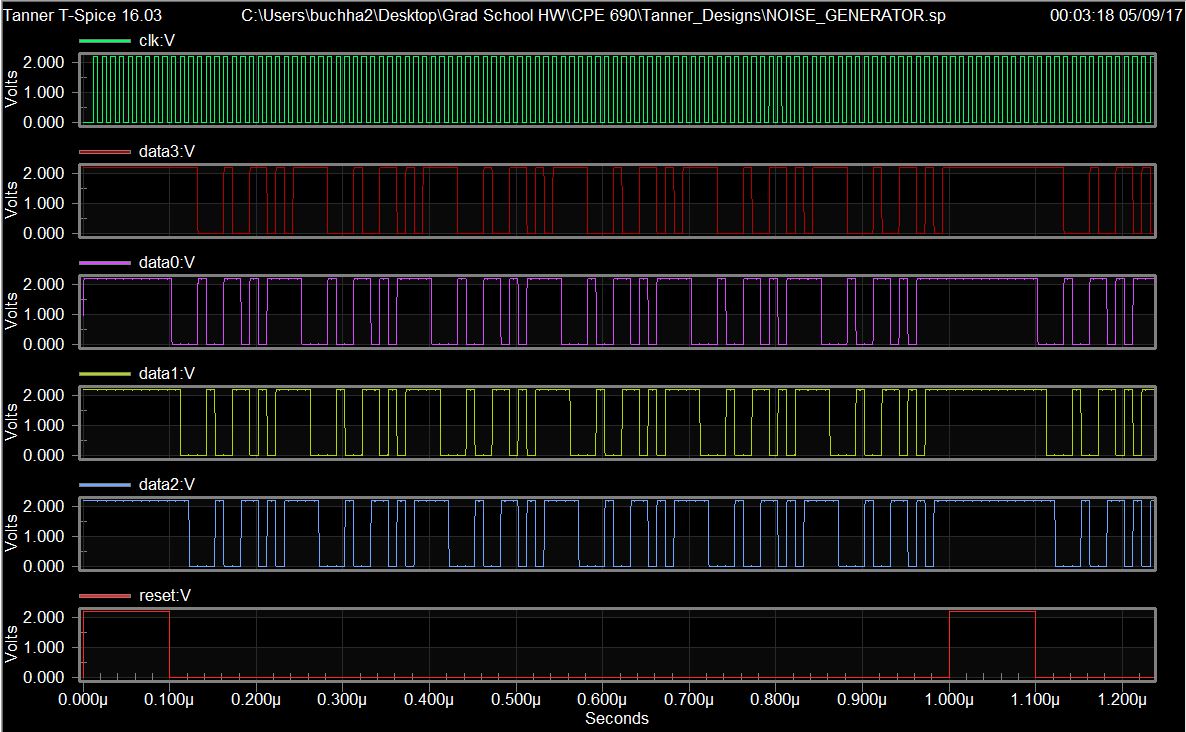
****

Figure 26 - 2.2V Inputs Simulation

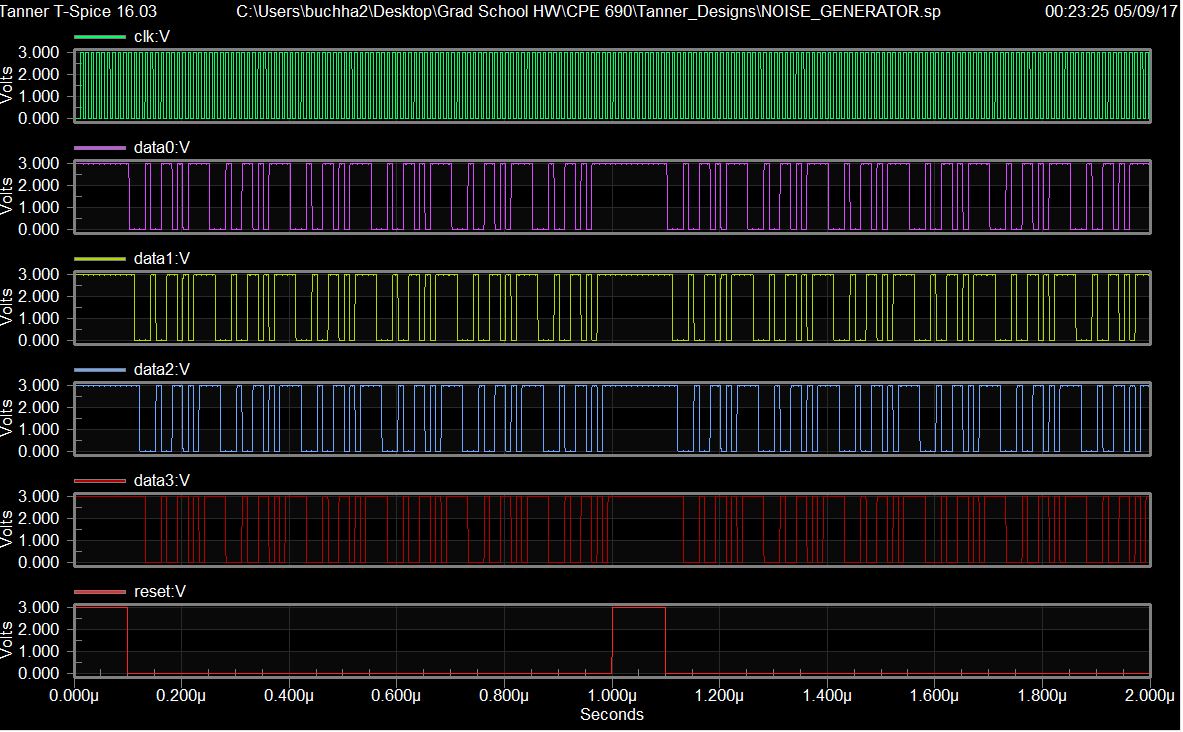
****

Figure 27 - 100°C Temperature Simulation

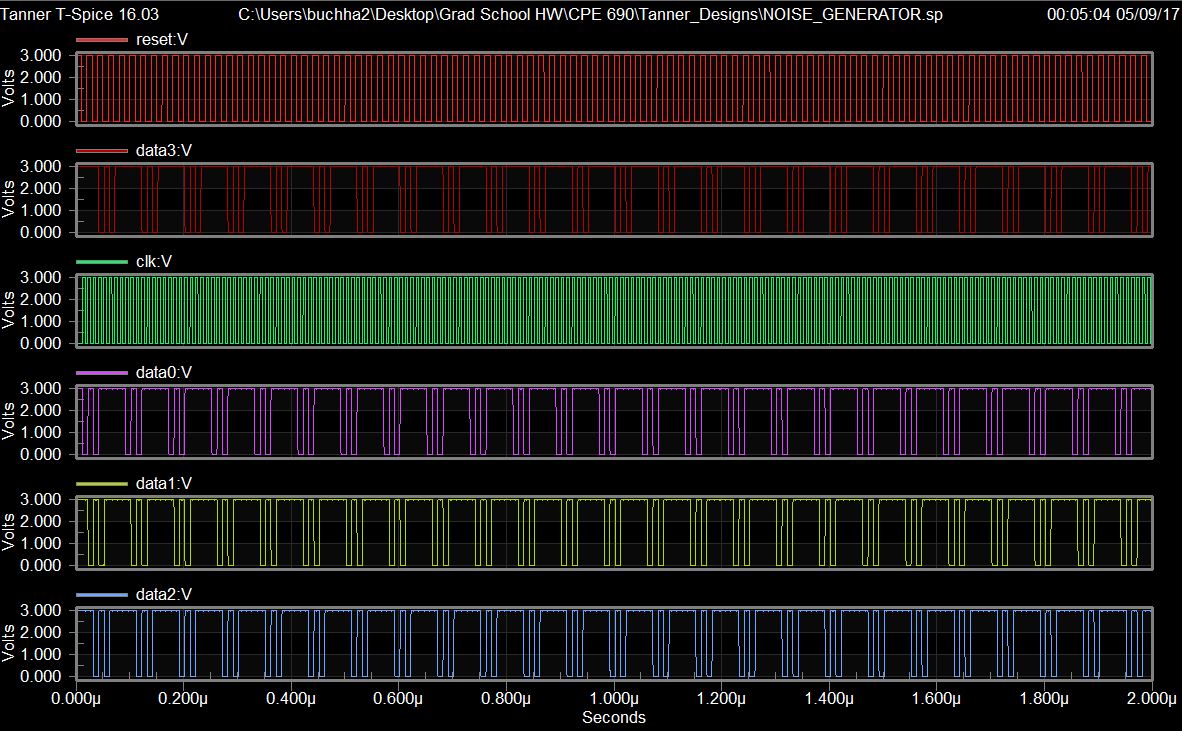
****

Figure 28 - Fast Reset Simulation

# Difficulties and Concluding Remarks

The most difficult part of this project was designing the D Flip-Flop. There was a period where I had a simulation error that took me several hours to debug, due to the sheer size of the device. Originally I had planned on only using two layers of metal, but I had to introduce a third layer because my layout was too cramped.

I also wish I had planned the dimensions a bit better; the length of the shift register is needlessly long, and the XOR2 and OR2 gates are awkwardly placed on the sides with metal rails sticking out.

If I had planned my layout better, I could have avoided simulation errors and made a much more compact design with less wasted space. Overall, however, it was an educational and rewarding experience.