

California State University, Northridge

Department of Electrical & Computer Engineering

## ECE 442L - Digital Electronics Lab

Fall 2023



### Lab 1

CMOS Ring Oscillation and Clock Generation

September 12, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

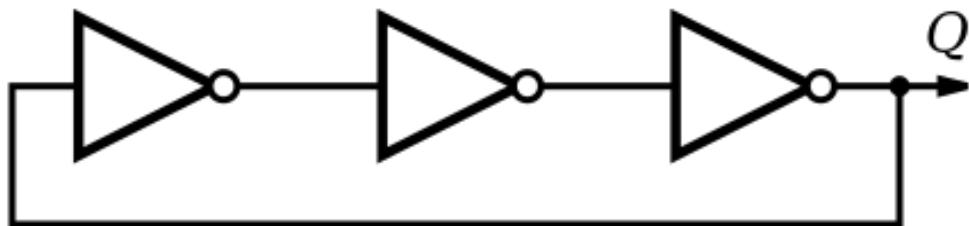
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion
6. References

## Abstract:

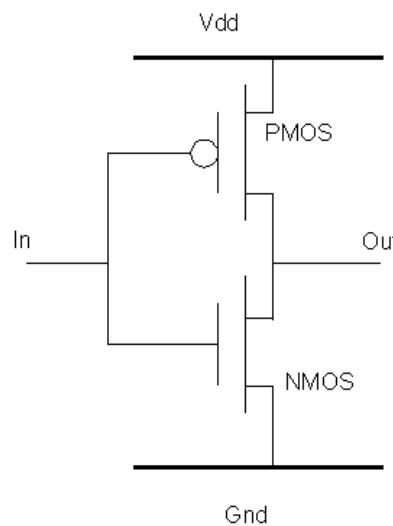
Computer clocks can be designed in a number of ways, one of which is chaining an odd number of inverters in a loop. This experiment aims at observing a basic ring oscillator operation. This was done by designing a ring oscillator using various inverter stages, building such oscillators, and observing if the frequencies were in line with expected values. The objective was met and observers left the lab with a basic understanding of ring oscillator operation.

## Theory:

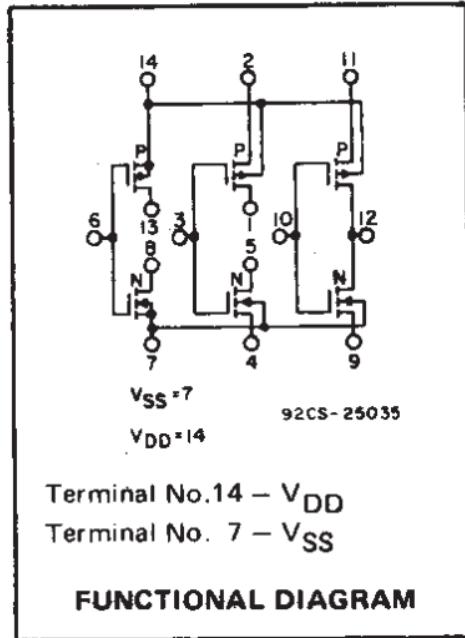
Ring oscillators behave as a clock switching between logic HIGH and LOW repeatedly. The oscillator is made up of inverters whose outputs are tied to other inverters inputs, this chain is made up of an odd number of links and the final inverter output is connected to the initial inverter input making the “ring”.



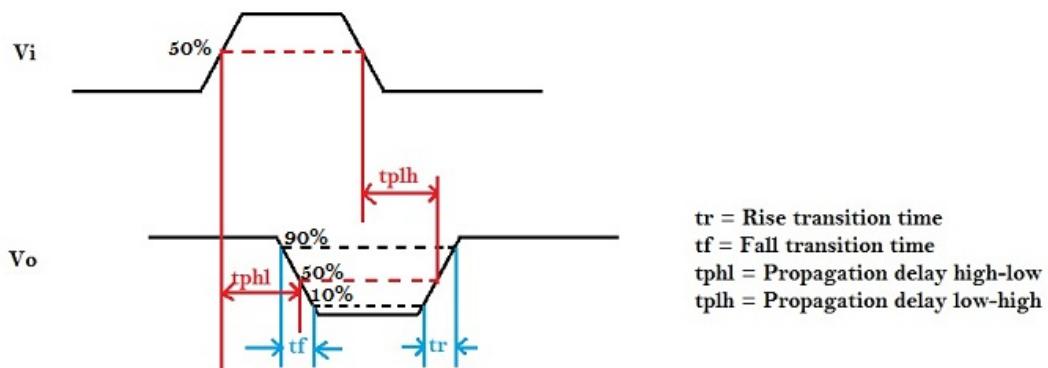
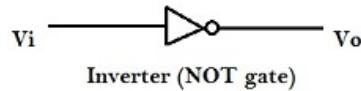
The invertors themselves are made up of CMOS technology transistors, the advantages of CMOS are simple structure, low power consumption, large noise tolerance and strong temperature stability. For the invertor a single NMOS and PMOS transistor connected in the way shown below.



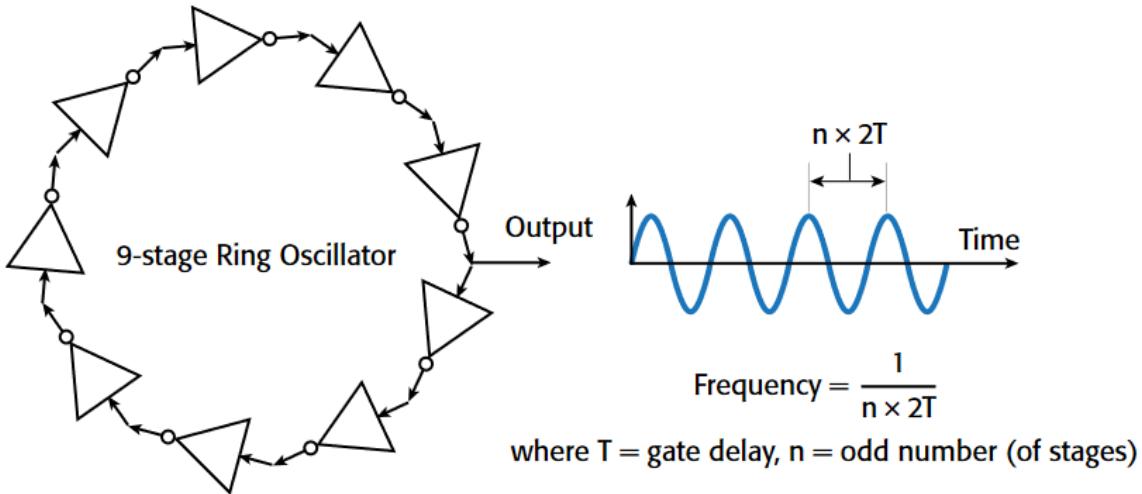
The chip used for the experiment is the Texas Instruments CD4007UB which is fitted with three pairs of transistors(each pair consisting of a single NMOS and PMOS).



The propagation delay is how long it takes a change in input to cause a change in output to appear, this value can be found by measuring from the half point of the input logic change to the half point of the output logic change as seen below as both t<sub>phl</sub> and t<sub>plh</sub>.



To get the average propagation delay we can get the average of t<sub>phl</sub> and t<sub>plh</sub>. According to the data sheet the typical propagation delay at a Vdd of 5V is 55ns. Assuming we are using seven inverters with the following formulas:



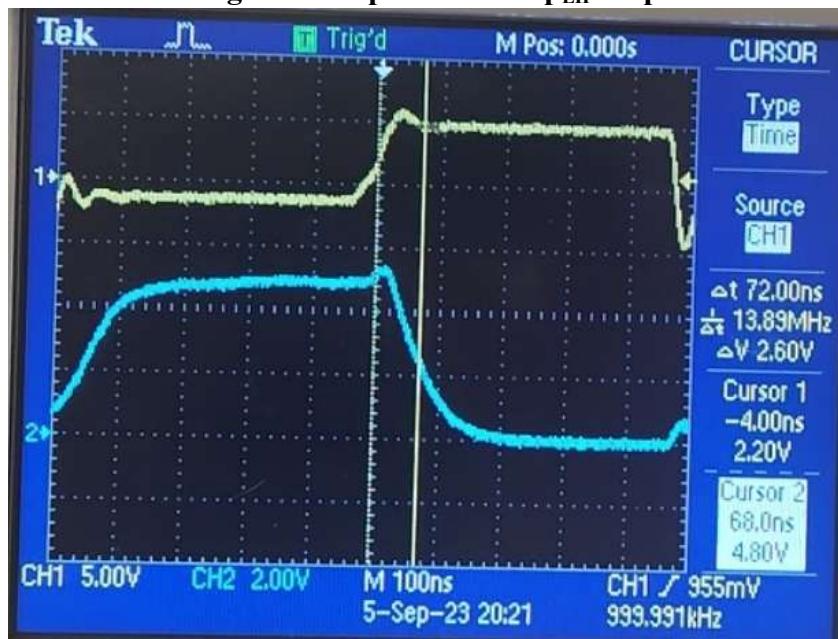
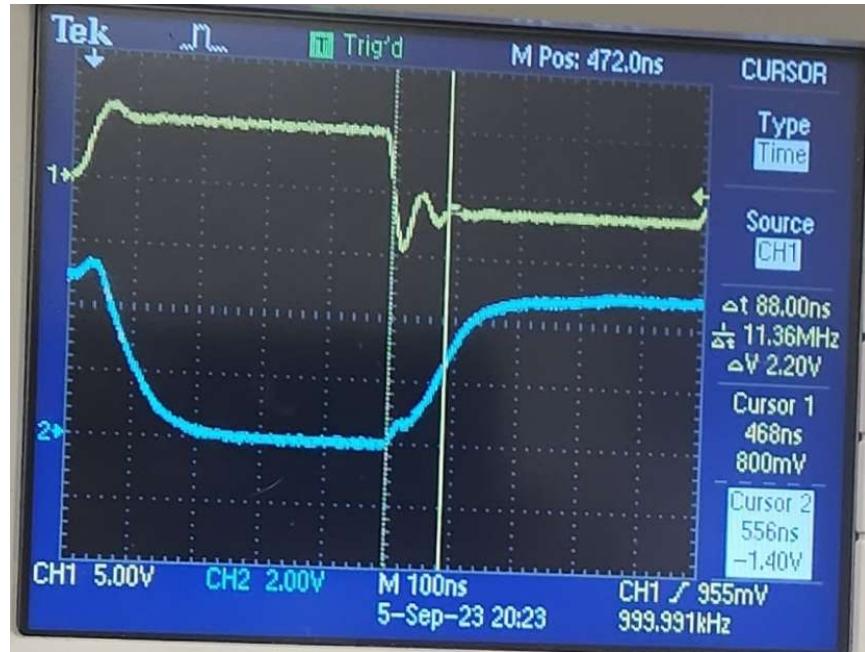
The frequency would come out to 1.299 MHz and the period at 770 ns. Is it possible to make a ring oscillator with a frequency of 20 GHz? To get a larger frequency a smaller number of inverters must be used, the smallest number of inverters possible is three. Calculating with propagation delay at 55ns and three inverters the frequency comes out to 3.03 MHz and 330 ns. Therefore it is not possible to get a frequency of 20 GHz.

### Experimental Results:

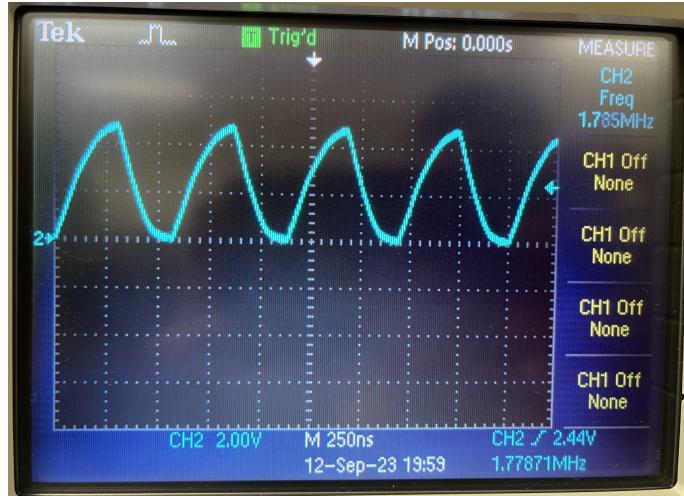
We first measured the propagation delay of our CD4007 chip by sending a pulse through one CMOS inverter and found our  $tp_{LH}$  and  $tp_{HL}$  to be 88ns and 72 ns respectively. For measuring  $tp_{LH}$  and  $tp_{HL}$ , we connected the input and output of the CMOS to an oscilloscope as shown in Figure 1 and Figure 2. As shown in Table 1,  $tp_{Avg}$  equals 80 ns, which was 30 ns above the expected value of the propagation delay from the datasheet.

$tp$ (from datasheet)	$tp_{LH}$ (experimental)	$tp_{HL}$ (experimental)	$tp_{Avg}$ $(tp_{LH}+tp_{HL})/2$
55 ns	88ns	72 ns	80 ns

Table 1: Propagation delay values



We configured our 7 ring oscillator using three CD4007 chips and measured the frequency as well as the period. We obtained a waveform with a frequency of 1.77 MHz and a period of 564.9 ns as shown in Figure 3. Using both of these values in our calculations, we obtained a propagation delay of 40.35 ns.

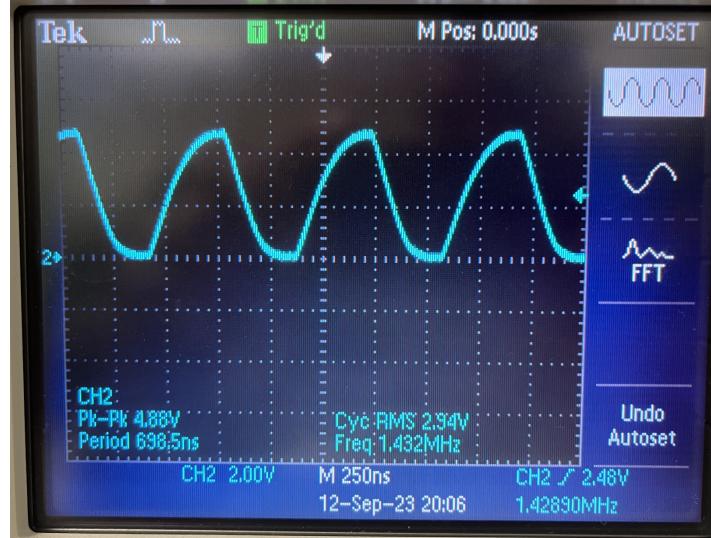


**Figure 3:** Experimental waveform of the 7-ring oscillator with  $f = 1.77\text{MHz}$

7-ring oscillator		
Propagation (tp)	Frequency (f)	Period (T)
40.35ns	1.77MHz	564.9ns

**Table 2: Experimental values from the 7-ring oscillator**

We then made our 9-ring oscillator and achieved a similar waveform as the 7-ring oscillator as shown in Figure 4. The frequency was 1.43 MHz and the period was approximately 700 ns, giving us a calculated propagation delay of 38.85 ns.



**Figure 4:** Experimental waveform of the 9-ring oscillator with  $f = 1.43\text{MHz}$ ,  $T=700\text{ns}$

9 ring oscillator		
Propagation (tp)	Frequency (f)	Period (T)
38.85ns	1.43MHz	700ns

**Table 3: Experimental values from the 9-ring oscillator**

**Applications:**

1. CMOS ring oscillators are used in wafer testing to test the effects of manufacturing processing on sets of hundreds of MOSFETS under high switching speeds[1]
2. Ring oscillators can be used for random number generation. Jitter in the output of the oscillator allows for random number generation in true random number generation (TRNG) devices. TRNG and physical unclonable functions (PUFs) are useful in secure system design [2]
3. Due to their sensitivity to temperature, ring oscillators can be used as on-chip thermal sensors. The frequency of oscillation is proportional to the temperature on the device and it is often displayed in a digital format [3]

**Conclusion:**

The ring oscillator experiment provided a practical understanding of CMOS based inverters and their associated propagation delays. While the CD4007 chip's measured propagation delay did not match the values from the Texas Instruments datasheet, the experiment successfully demonstrated the relationship between oscillation frequency and ring count. The observed deviations between theoretical values and physical measurements demonstrate the importance of real world experimentation.

California State University, Northridge

Department of Electrical & Computer Engineering

# ECE 442L - Digital Electronics Lab

Fall 2023



## Lab 2

CMOS Inverter Voltage Transfer Characteristics(VTC)

September 19, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

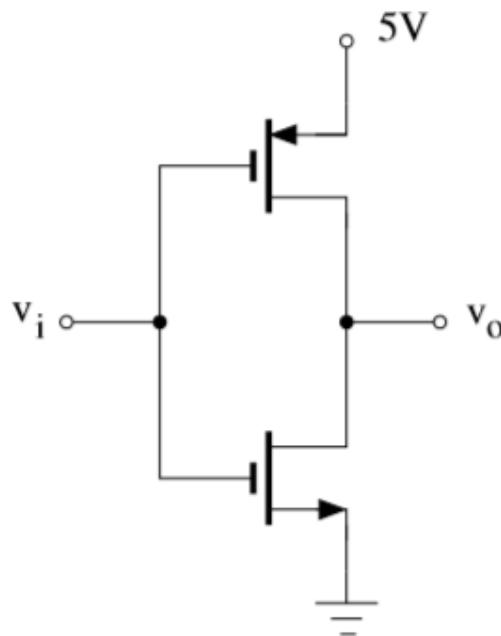
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## **Abstract:**

The Inverter is a fundamental logic gate among the three basic logic gates which can be used to make any boolean circuit. This experiment aims at observing the Voltage Transfer Characteristics(VTC) of a CMOS inverter. This was done by modeling a CMOS inverter in SPICE using the Texas Instruments CD4007 model, plotting its VTC and building the circuit with the real component to compare the characteristics. The objective was met and observers left with a basic understanding of CMOS inverter voltage characteristics, and the limitation of models.

## **Theory:**

Paired with the AND gate, and the OR gate the inverter can implement any binary function. This simple device takes a logic input and outputs the opposite value. The inverter can be designed using simply two transistors, a PMOS and NMOS configured in such fashion:

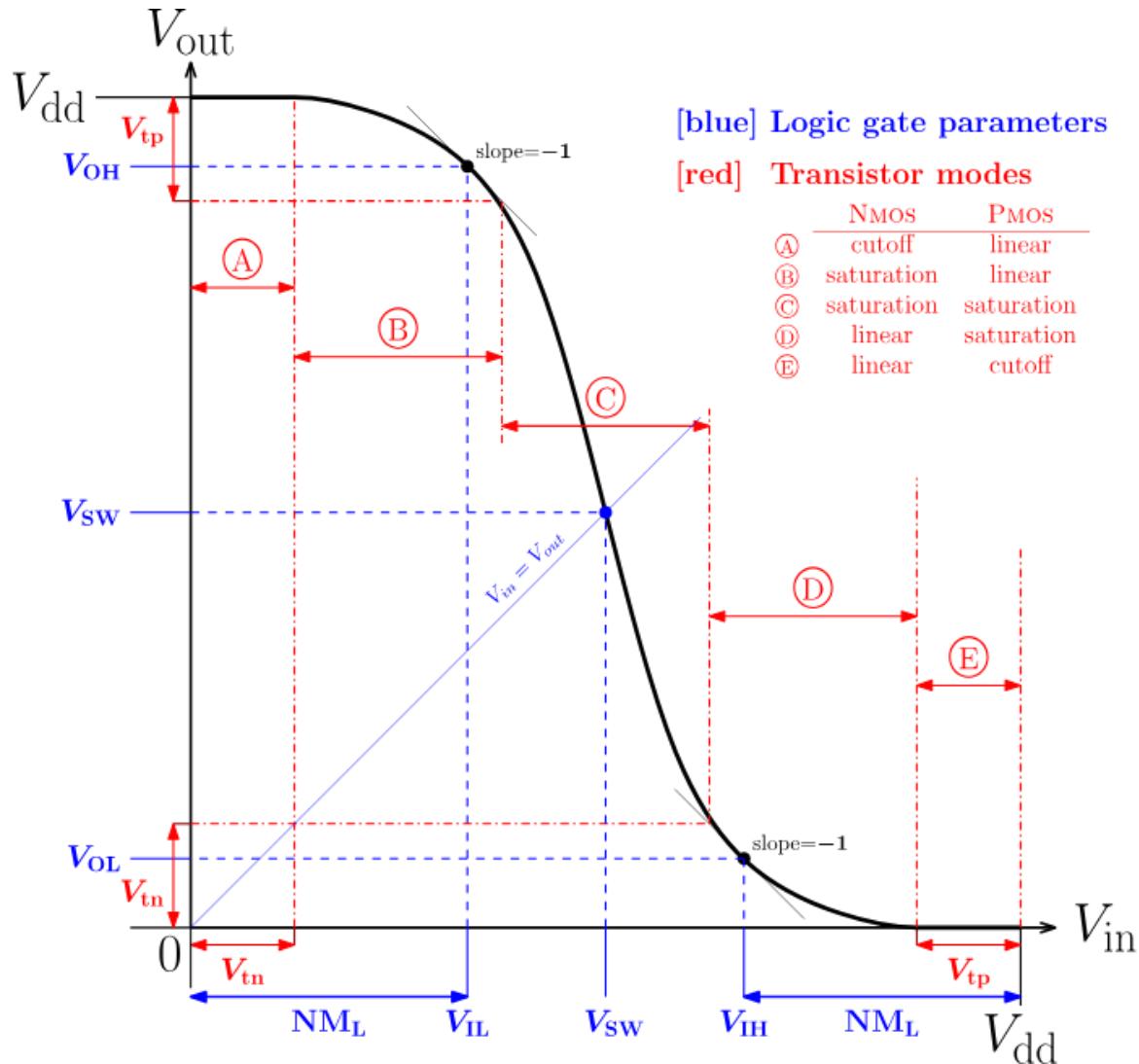


The NMOS is located on the bottom with source at ground and the PMOS is located on top with its drain at the source, 5V in this case. We will observe the two states of the inverter:

1. Looking at the schematic, assuming the input  $v_i$  is low, the NMOS should be in the Cutoff region and the PMOS should be in TRIODE. This results in the PMOS acting like a resistor between the output and source voltage, therefore the output would have the same value as the source or high.

2. Now if the input is high this will cause the PMOS to go into cutoff and the NMOS to go into TRIODE. As a result the NMOS will behave like a resistor between output and ground making the output low.

If the input voltage were to be ramped from zero to source or VDD, then the input vs output voltage graph would look like so:



The inverter output does not instantaneously flip as can be seen in the VTC, the switching point is where a line of slope one from the origin intercepts the VTC curve( $V_{sw}$  in diagram also called  $V_m$ ) if the curve is symmetrical then the  $V_m = V_{dd}/2$  and it is called balanced. Two identical size NMOS and PMOS transistors would not likely be balanced if used for an inverter. This is due to the carriers in a PMOS being holes which have a lower electron mobility than the NMOS electron carriers. If a transistor were to be made with identical size NMOS and PMOS transistors the the  $V_m$  would be shifted away from

the middle. To get an ideal balanced inverter adjustments need to be made such as making the PMOS wider so the currents even out.

### Experimental Results:

We first designed a CMOS inverter in PSPICE using ideal models of a PMOS and CMOS with a width of 1  $\mu\text{m}$ .

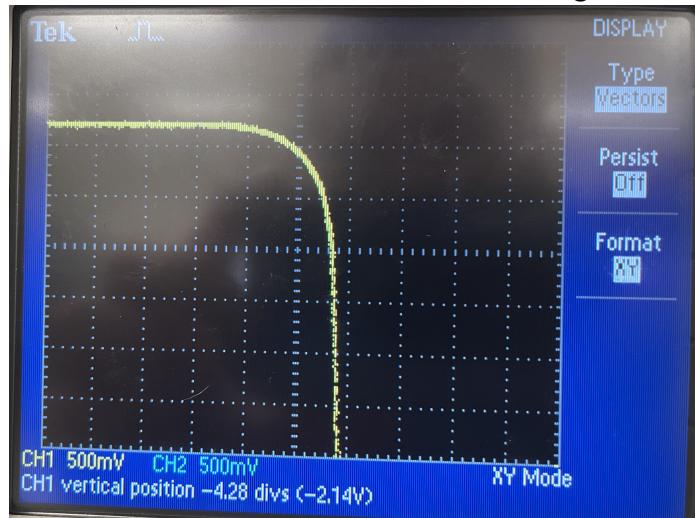
Once the CMOS inverter was created, we generated a VTC curve by sweeping the voltage from 0-5 volts, with  $\text{VDD} = 5$  volts.

From Table 1, we found the switching threshold ( $\text{Vm}$ ) to be 2.5V by calculating  $\text{VDD}/2$ . We used the low input ( $\text{V}_{\text{IL}}$ ) and high input voltage ( $\text{V}_{\text{IH}}$ ) values to determine our threshold voltage ( $\text{Vt}$ ) = 0.1V. Since both transistors are matched, the noise margin high value ( $\text{NM}_H$ ) is equal to the noise margin low ( $\text{NM}_L$ ), hence both are equal to 1.9V.

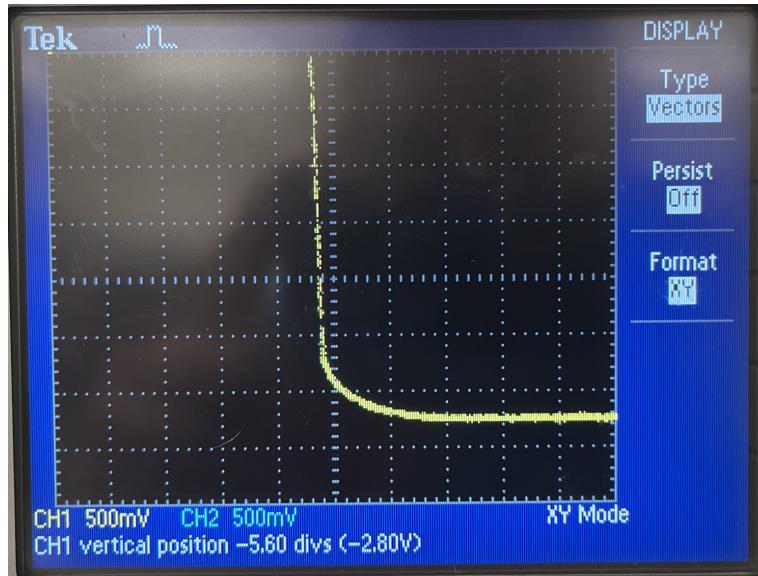
$\text{Vm}$	$\text{V}_{\text{IL}}$	$\text{V}_{\text{IH}}$	$\text{NM}_H=\text{NM}_L$	$\text{Vt}$
2.5 V	1.9 V	3.1 V	1.9 V	0.1V

**Table 1: Switching Threshold, Gain, and Noise margin simulation values**

To achieve experimental results, we built our own CMOS inverter using our CD4007 chip and created a VTC curve as shown in Figure 1 and 2.



**Figure 1: Experimental VTC Curve, at  $\text{V}_{\text{IL}}= 2.14\text{V}$**



**Figure 2: Experimental VTC Curve, at  $V_{IH} = 2.8V$**

With our VTC curve, we were able to determine the low input ( $V_{IL}$ ) and high input voltages ( $V_{IH}$ ). For finding the threshold voltage ( $V_t$ ), a voltage sweep was done manually while observing the change in voltage through our multimeter. We increased the voltage until the threshold voltage occurred at 1.62 volts as shown in figure 3.



**Figure 3: Experimental value of  $V_t = 1.62V$**

Once we obtained the threshold voltage, the noise margin high and low values were calculated and found to be 2.28V. All experimental values are shown in Table 2.

<b>V<sub>m</sub></b>	<b>V<sub>IL</sub></b>	<b>V<sub>IH</sub></b>	<b>NM<sub>H</sub>=NM<sub>L</sub></b>	<b>V<sub>t</sub></b>
2.60V	2.14V	2.80V	2.28V	1.62V

**Table 2: Switching Threshold, Gain, and Noise margin experimental values**

### **Applications:**

CMOS inverters are building blocks in digital ICs. The voltage transfer characteristics of a CMOS inverter relate the input voltage to the output voltage. Understanding the characteristics of a CMOS allows us to understand the behavior of them in the following applications.

1. Oscillators. CMOS inverters can be used to create clocks through oscillation and understanding their characteristics is fundamental in determining their frequency.
2. Digital logic signals. One of the main applications of CMOS inverters is to invert digital logic signals. When an input comes in high (Vcc) the output comes out low (0V) and vice versa. The characteristics allow us to understand how and at what V<sub>in</sub> levels signals are inverted.
3. Amplifiers. In analog circuits the CMOS inverter can be used as an amplifier in the linear region of the characteristic graph.

### **Conclusion:**

After constructing the simulated CMOS circuit in PSPICE and analyzing it we determined that the values were significantly different from the physical CMOS. While V<sub>m</sub>, V<sub>IL</sub>, V<sub>IH</sub>, and NM<sub>H</sub> = NM<sub>L</sub> were within a reasonable range, V<sub>t</sub> was vastly different with simulated V<sub>t</sub> at .1V and experimental V<sub>t</sub> at 1.62V. We concluded that the difference between the V<sub>t</sub> values could be explained by the fact that the simulated CMOS was ideal while the experimental one wasn't. This project provided valuable insights into the function of CMOS inverters and demonstrated how important real world experimentation is.

California State University, Northridge

Department of Electrical & Computer Engineering

## ECE 442L - Digital Electronics Lab

Fall 2023



### Lab 3

Design of a CMOS 2 Input NAND Gate

September 26, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

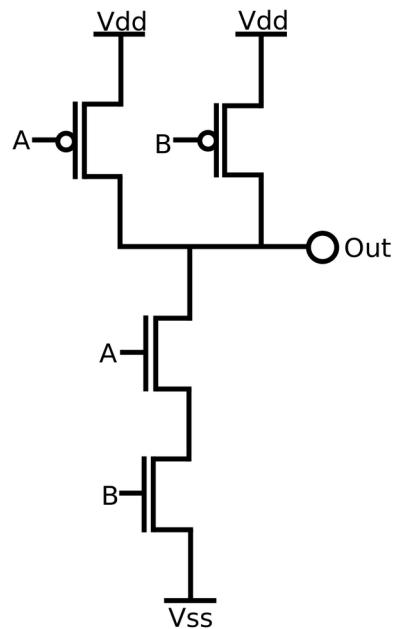
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## **Abstract:**

The CMOS 2 input NAND gate is a fundamental logic gate used as a building block for more advanced digital devices. This experiment aimed at designing and building a CMOS NAND gate. This was done by first designing and simulating the circuit of the NAND gate using PSPICE checking to see logical wave forms. Then the circuit was built using the TI CD4007 CMOS chip. The objective was met and observers left with an understanding of CMOS NAND behavior, and implementation.

## **Theory:**

The CMOS NAND gate, an acronym for Complementary Metal-Oxide-Semiconductor NAND gate, serves as a fundamental digital logic gate designed for executing the logical operations associated with the Boolean NAND function. Within the context of a 2-input CMOS NAND gate, denoted by input signals A and B, the gate yields a low output (logical 0) exclusively when both A and B register as high (logical 1). For all other combinations of input states, the output maintains a high status (logical 1). In scenarios where both inputs (A and B) are set to logic 1, NMOS transistors become conductive, establishing a low-resistance path to ground, while the PMOS transistors remain in the off state. This action pulls the output node to ground, resulting in a logical 0 output. Conversely, if either or both inputs are set to logic 0, the corresponding transistors within the complementary pair deactivate, causing the output node to be pulled up to the supply voltage, thereby yielding a logical 1 output.



**Figure 1: CMOS NAND gate diagram**

The CMOS NAND gate's strength lies in its inherent capability to provide complemented outputs (inverted outputs), rendering it a versatile foundation for the construction of diverse digital circuits. The widespread adoption of CMOS technology in integrated circuits is attributed to its advantages, including low power consumption and excellent noise margins, establishing the 2-input CMOS NAND gate as an indispensable component in contemporary digital electronics.

## Experimental Results:

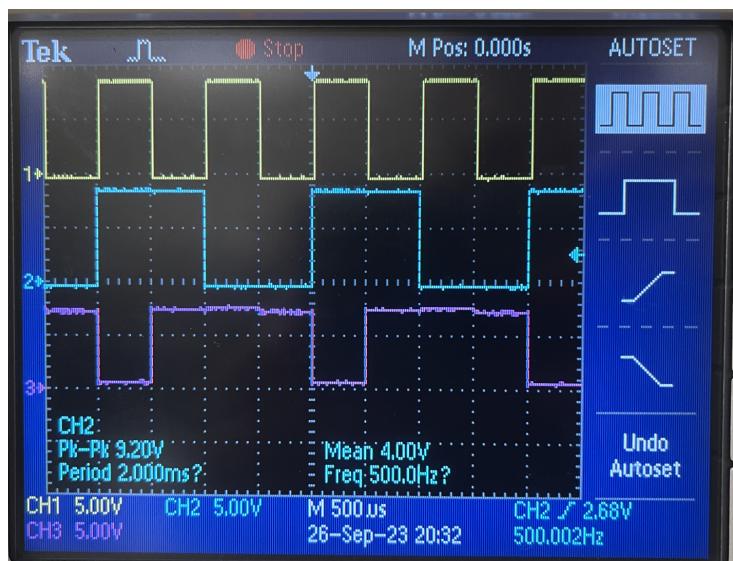
We designed a 2-input CMOS NAND based on our CMOS inverter as shown in Figure 1. The p-channel transistors are connected in parallel between VDD and the output terminal. The n-channel transistors are connected in series between the output terminal and ground.

(PSPICE CMOS NAND SCREENSHOT)

The truth table for a 2-input NAND gate states that both inputs (A and B) must have a logic 1 simultaneously in order for the output to be logic 0. The waveform simulated in figure 2 shows when both inputs are in logic 1. As a result, The output is in the bottom plot and in logic 0, validating the bottom row of the truth table for a 2-input NAND CMOS circuit.

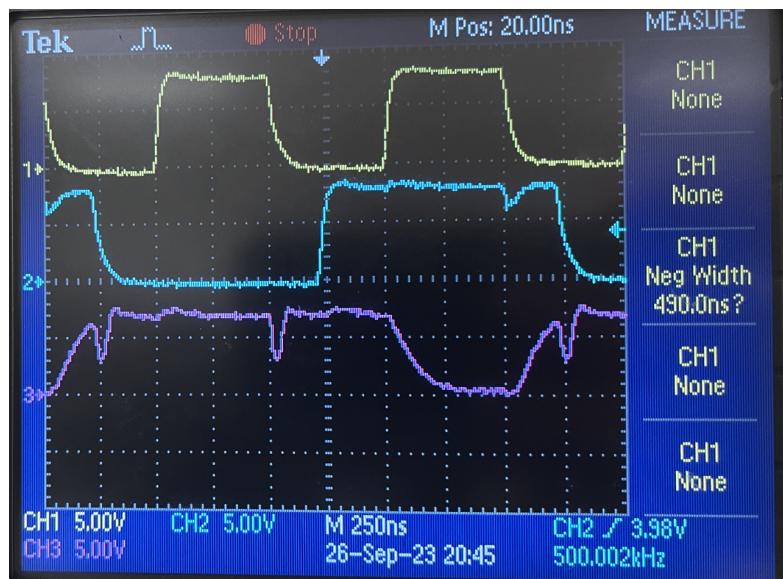
(PSPICE WAVEFORM SCREENSHOT)

After simulation, we implemented our own 2-input NAND circuit using our CD4007 chip. With VDD = 5 volts, we generated a separate pulse for input A and input B with frequencies of 2 kHz and 1 kHz respectively. We visualized our NAND circuit via the oscilloscope as shown in figure 3. Our waveform confirms when both inputs are high, the output will be low.



**Figure 3 Experimental Waveform of CMOS 2-input NAND circuit**

When we increase the frequencies to input A and B by 1MHz and 500kHz respectively, our transistors can no longer sustain a consistent waveform as shown in figure 4.



**Figure 4 Experimental Waveform of CMOS 2-input NAND circuit**

### **Applications:**

CMOS NAND gates are fundamental in digital circuit design. Some of their applications include.

1. Clock generation. When you incorporate other components along with the CMOS NAND you can generate clock signals.
2. Memory elements. CMOS NAND gates can be used in the creation of memory elements like flip flops. Flip flops and other memory elements are essential in sequential circuits.
3. Logic functions. With just NAND gates it is possible to create many other logic gates such as NOT, AND, and OR.

### **Conclusion:**

In conclusion, after the construction of both simulated and physical circuits we were able to input two signals into the NAND gate and see the correct output. Both input signals were square waves of 5V amplitude but their frequencies differed, wave one was at 500 Hz while the second one was at 1 kHz frequency. The output of the nand gate was consistent with the theoretical gate truth table. Our simulated circuit ran successfully but our physical circuit was initially outputting the incorrect signal due to the 2 square waves not being aligned with each other. After offsetting one of the square waves the output signal was correct. This project provided valuable insight into one of the building blocks of digital logic and how they can be implemented using CMOS technology.

California State University, Northridge

Department of Electrical & Computer Engineering

# ECE 442L - Digital Electronics Lab

Fall 2023



## Lab 4

### Transmission Gate

October 3, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

## Table of Contents

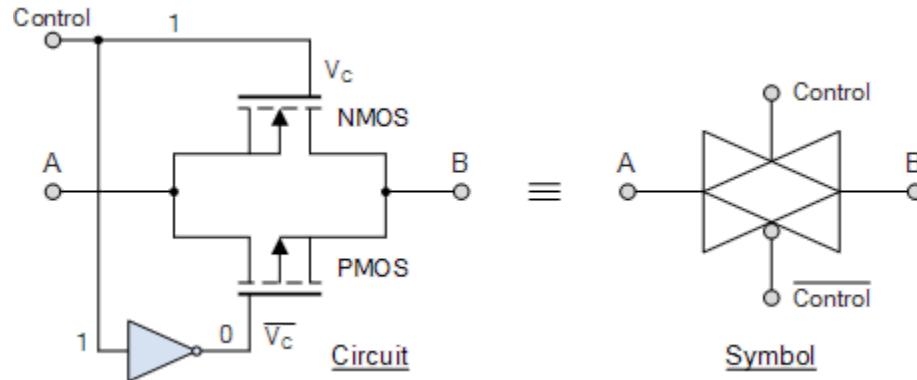
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## Abstract:

The Transmission Gate is a useful device in acting as a gate however without the voltage limitations of NMOS and PMOS transistors on their own. The experiment aimed at designing, simulating and building a Transmission Gate, then using transmission gates to make a quantizer. This was done by simulating the transmission gate and quantizer in PSPICE and then attempting to build said devices. Observers were unable to build a practical transmission gate due to unknown reasons. Although real devices were not built observers were able to learn transmission gate and quantizer function through the troubleshooting process.

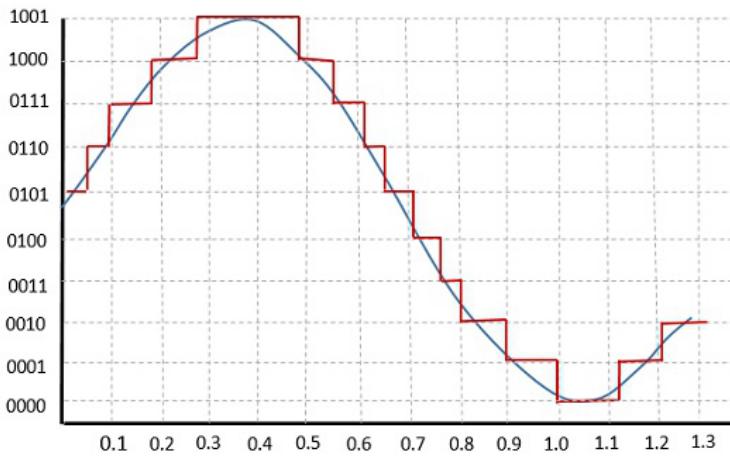
## Theory:

The Transmission Gate that allows for the functionality of a gate without the drawbacks of the NMOS or PMOS alone. When using an NMOS as a gate it is possible to pull the voltage to ground, however it is unable to pull the voltage up to the source. While the PMOS is the opposite in that it can pull up to a full HIGH but fails to pull down to ground. Using both transistors together allows for the full voltage range to pass through the gate.



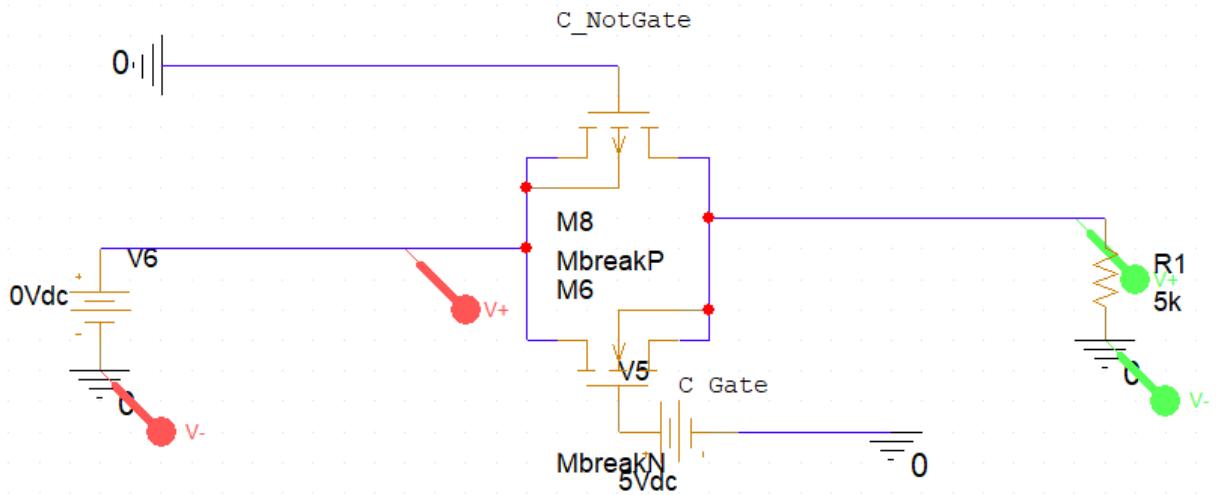
The transmission gate is put together as can be seen in the figure above. It should be noted that to turn on an NMOS the gate should be HIGH, and to turn on a PMOS the gate should be LOW. Therefore to control the whole transmission gate together the control signal should be inverted and to control one of the transistors (trace the control signal in the figure). Now with this layout a gate that passes voltage from A to B or vice versa is functional pulling from 0 all the way to VDD.

This is useful in the case of an analog to digital converter, by clocking the Transmission gate control it is possible to pass the voltage to charge a capacitor when the gate is on which will hold that charge when the gate is off converting the input signal to a quantized signal like so:



## Experimental Results:

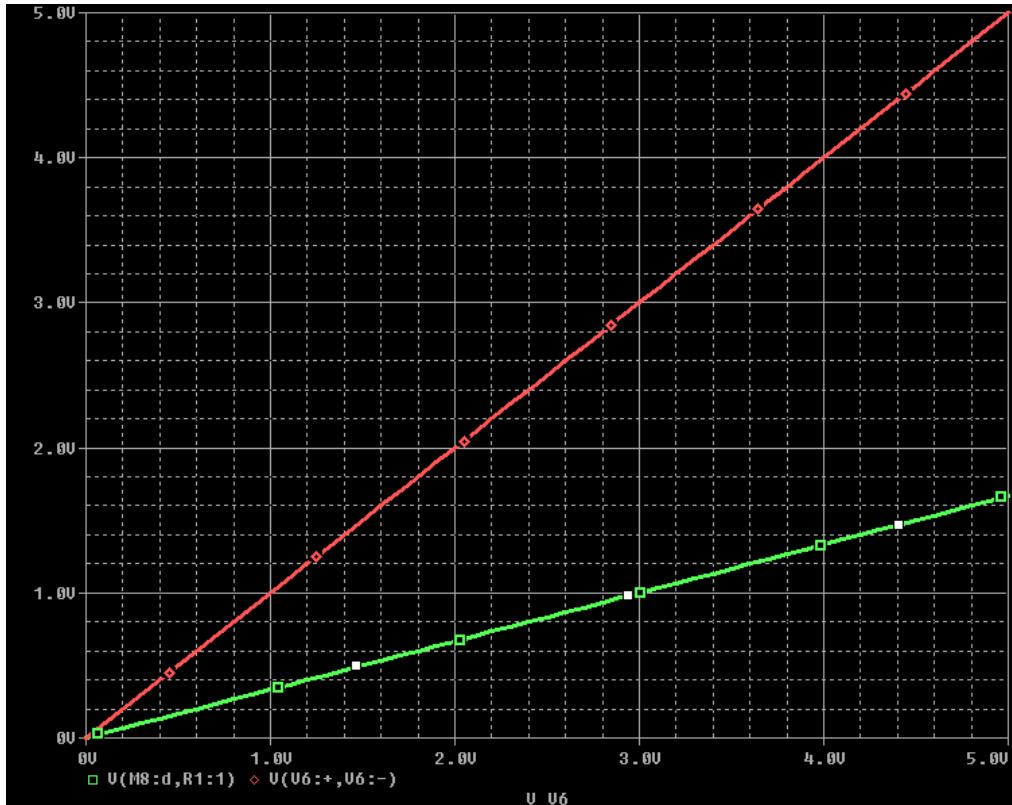
Prior to implementing a physical circuit of a CMOS transmission gate, an initial step involved creating a simulated circuit using PSPICE. This simulation aimed to ascertain optimal timing and illustrate the functionality of the transmission gate. In Figure 1, the PSPICE circuit features the T-Gate designated for testing. Positioned above is a PMOS for the clock bar gate, linked to ground, while below is an NMOS with the C gate connected to a 5 V source. This ensures that both transistors will be on when an input is injected to the circuit. Additionally, a 5k resistance is incorporated at the output of the T-gate.



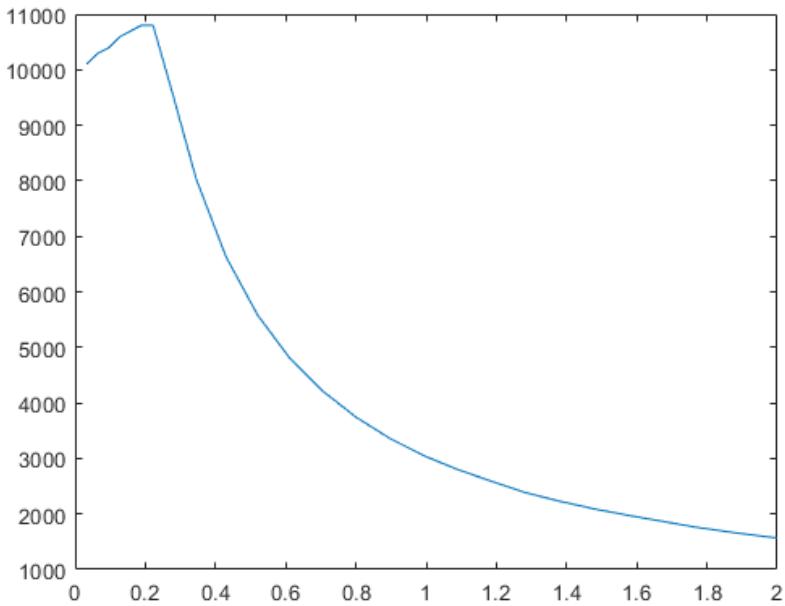
**Figure: 1 PSPICE transmission gate circuit with a 5k resistor at the output**

A ramp input is applied to the input of the transmission gate and is simulated in the PSPICE waveform shown in Figure 2. This data was then exported to MATLAB where the equivalent resistance of the T-gate ( $R_{tg}$ ) was plotted in Figure 3. The highest  $R_{tg}$  peaked at approximately

$10.9 \text{ k}\Omega$  at 0.2 V. The average  $R_{tg}$  was calculated to be  $10.4 \text{ k}\Omega$ . If the Resistance on the output were to increase, the change in the resistance on the transmission gate could possibly introduce more signal attenuation which affects the voltage levels at the output. This is because the output is influenced by the parallel combination of  $R_{tg}$  and the resistance of the pull-up and pull-down networks. Nevertheless, the average resistance recorded in the simulation will generally minimize power dissipation while not sacrificing its fast signal transition.



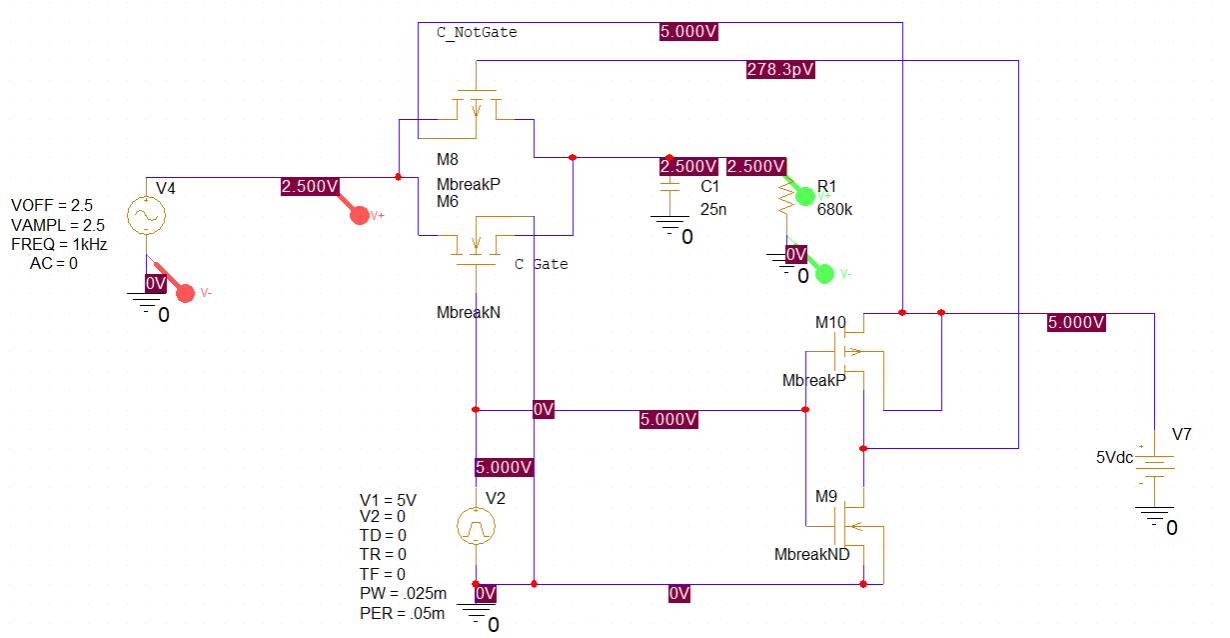
**Figure 2: PSPICE waveform of  $Vi$  (red) and  $Vo$  (green) with relation to voltage ramp of 0-5V**



**Figure 3: Rtg Values with a 5k Resistor at the output of the transmission gate.**

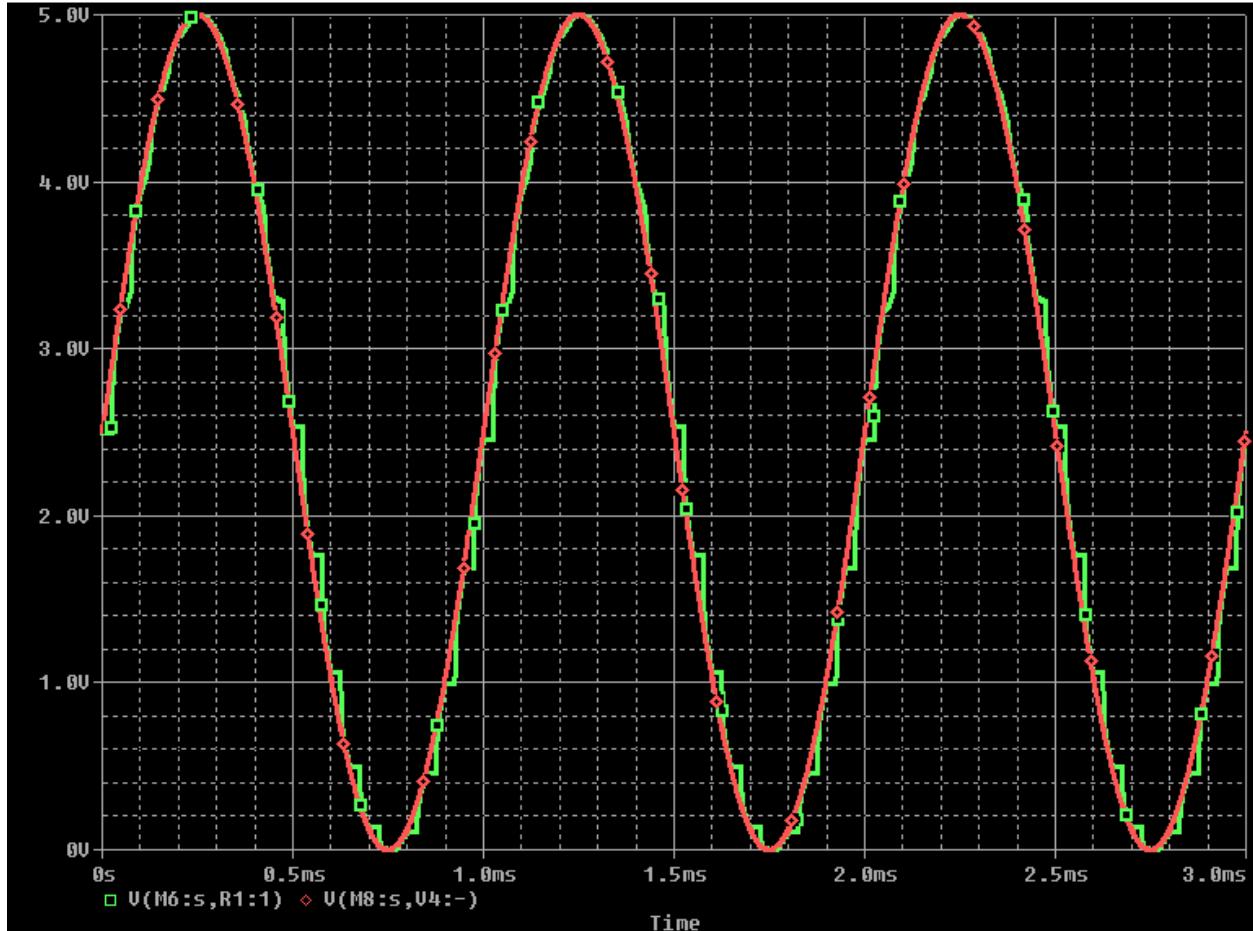
$$R_{tg} = 10.4 \text{ k}\Omega$$

The illustrated PSPICE circuit in Figure 4 represents a sample and hold circuit employing a CMOS T-gate utilizing 1  $\mu\text{m}$  transistor models. A sinusoidal voltage source, set with an amplitude of 2.5 V at a frequency of 1 kHz, was positioned at the input of the T-gate. Also, a pulse generator was installed onto the circuit with an input voltage of 5V and at a frequency of 20 kHz. There were issues with the circuit not being able to hold when analyzing the simulated waveforms measuring the input and output of the T-Gate.



**Figure 4: Sample and Hold transmission gate PSPICE circuit**

The solution to all problems was to increase the output resistance to 680k and also increase the output capacitance to 25 nF. The capacitance value before was too low, effectively charging the capacitor and not allowing the sample phase to hold. With the new output resistance and capacitance sizes, the voltage that is stored in the capacitor ensures that the sampled voltage maintains a smooth transition from sampling to holding, as shown in figure 5.



**Figure 5: Waveform of Sample and Hold transmission gate circuit**

### Applications:

CMOS Transmission gates, also known as pass gates, are versatile components in digital circuit design. Here are some key applications.

1. Bidirectional data transmission. Transmission gates allow the driving of signals in both directions, making them suitable for data buses in processors and memory devices that require bidirectional operation. This allows for effective data flow between different parts of a circuit or between different circuits.
2. Sample and Hold Circuits. Transmission gates are used in sample and hold circuits to capture and hold an analog value for some time. This application is vital in ADC, where a stable signal level is required for accurate conversion while sampling.
3. Multiplexers and Demultiplexers. Transmission gates are very useful in signal processing and communication systems. They are often found in mux and demux, enabling one signal to be connected to multiple outputs or combining multiple inputs to a single output line.

**Conclusion:**

In conclusion, after the construction of the simulated circuit we observed correct behavior from the circuit and found the simulated version of RTG which was  $10.4\text{ k}\Omega$ . Following this we implemented the sample and hold using the simulated transmission gate and were able to get a digital representation of an analog signal as seen in Figure 5. For our experimental circuitry, although we were able to find the value of RTG for the transmission gate made with the CD4007 chip we could not complete the sample and hold operation. Even without a functional experimental circuit this lab taught us about the function of transmission gates as well as their implementation.

California State University, Northridge

Department of Electrical & Computer Engineering

## ECE 442L - Digital Electronics Lab

Fall 2023



### Lab 5 & 6

CMOS D-latch and NAND based S-R Latch

October 24, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## Abstract:

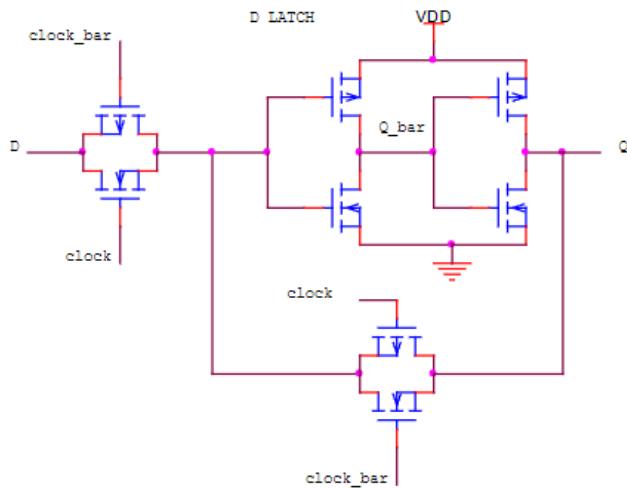
The D-latch and SR-latch are key components in memory storage. This experiment was aimed at designing, simulating and building a D-latch and SR-latch. This was done by first simulating the latches within PSPICE and getting an expected waveform from the circuit simulation. Then the physical circuit was attempted to be built using the TI CD4007 chip, although issues occurred with the D-latch where simple logic inputs behaved as expected clocked inputs didn't. The SR-latch was a success and behaved as expected. The objective was mostly met and observers gained understanding of CMOS latch construction.

## Theory:

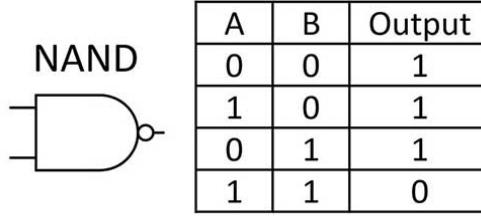
The D-latch is a fundamental digital circuit element used for storing a single bit of information. The CMOS D-latch is designed using complementary pairs of PMOS and NMOS transistors. The latch operates based on a clock signal, enabling or disabling the transfer of data to the storage node based on the clock state. D-latches have no “illegal signals” and their truth tables are as so:

E	D	Q	$\bar{Q}$
0	0	latch	latch
0	1	latch	latch
1	0	0	1
1	1	1	0

D-latches are level triggered as opposed to D flip-flops which are edge triggered. The circuit can be built using two Transmission gates as well two inverters.



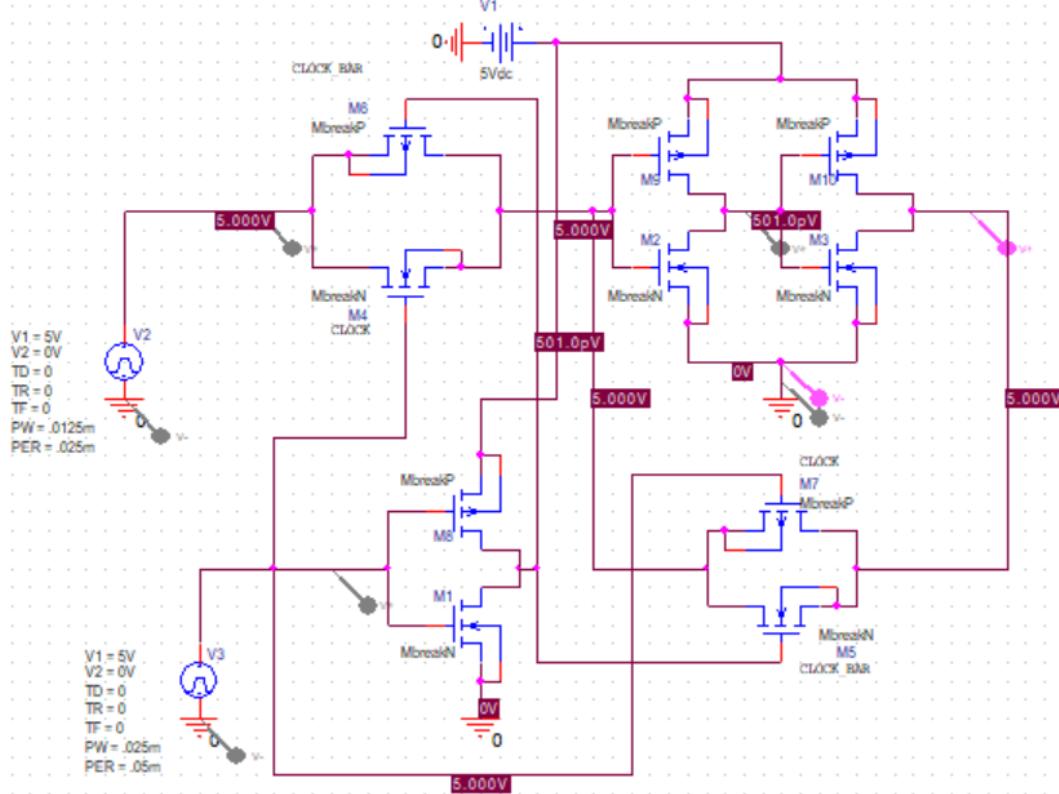
A CMOS SR latch constructed from NAND gates typically consists of two cross-coupled NAND gates. Each NAND gate comprises both PMOS (p-type metal-oxide-semiconductor) and NMOS (n-type metal-oxide-semiconductor) transistors. The cross-coupling of the two NAND gates forms a positive feedback loop, allowing the latch to store and maintain a binary state.



CMOS SR latches made of NAND gates have low static power consumption when not undergoing state transitions. Dynamic power consumption occurs during transitions between set and reset states.

### Experimental Results:

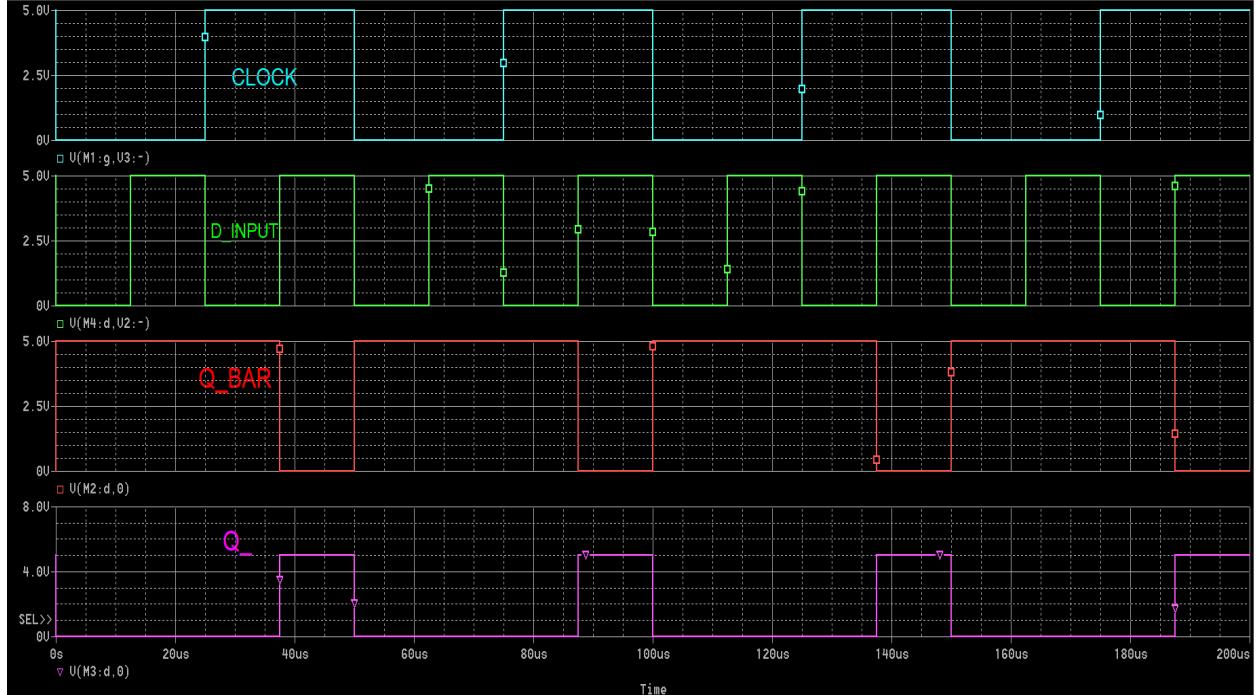
The CMOS D-latch was designed using inverters and transmission gates. The inverters were sized for a switching threshold of  $V_{dd}/2$  and the transistors of the transmission gates to be of equal size (Matched case). In figure 1, D-input is a pulse generator of 0-5V and a frequency of 40 kHz. The clock is a pulse generator as well of 0-5V, and with a frequency of 20 kHz.



**Figure 1: D-Latch PSPICE circuit with  $VDD=5V$ ,  $VPULSE=5V @40kHz$  (1/0.025ms),  $VCLOCK=5V @20Khz$  (1/0.05ms)**

The waveforms simulated in figure 2 show the successful operation of the D-latch. When the clock signal changes from one state to another, the D-latch captures and holds the input data. For example, when the clock and D-input are both high at around 38 us, the Q output rises high as

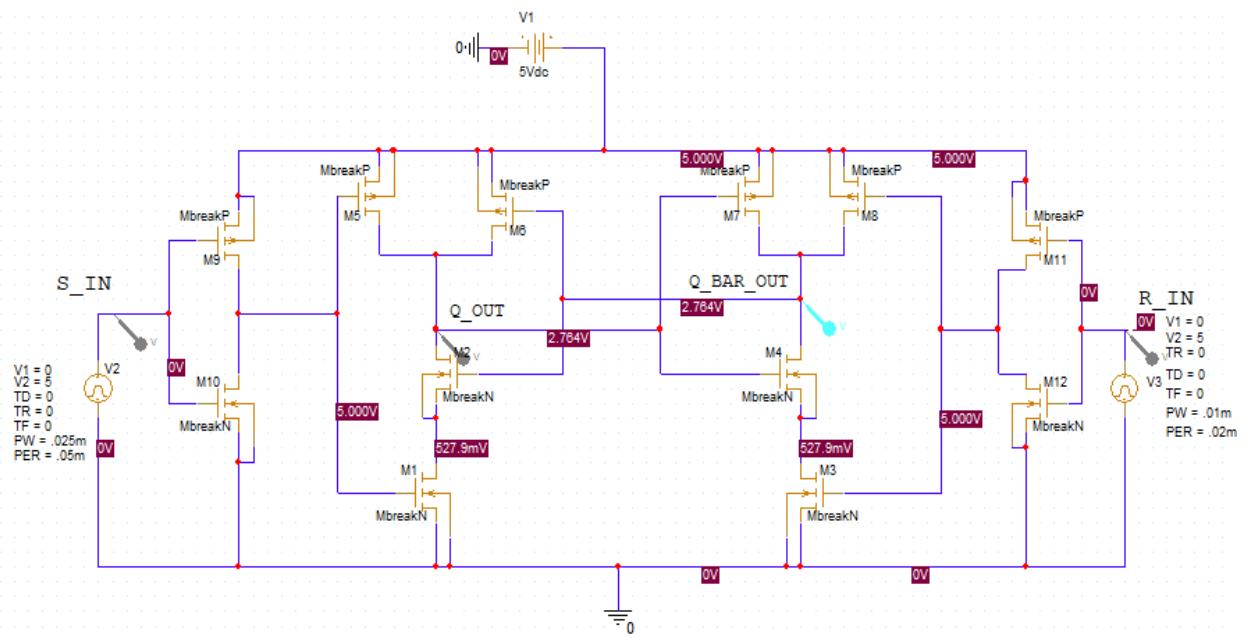
well. Then, as the clock and D-input go low, the stored one resets to zero and holds this value until the input and clock input simultaneously go high. The ability to hold this value ensures the PSPICE design is reliable for comparing our results with the physical circuit when measuring the inputs and outputs on the oscilloscope.



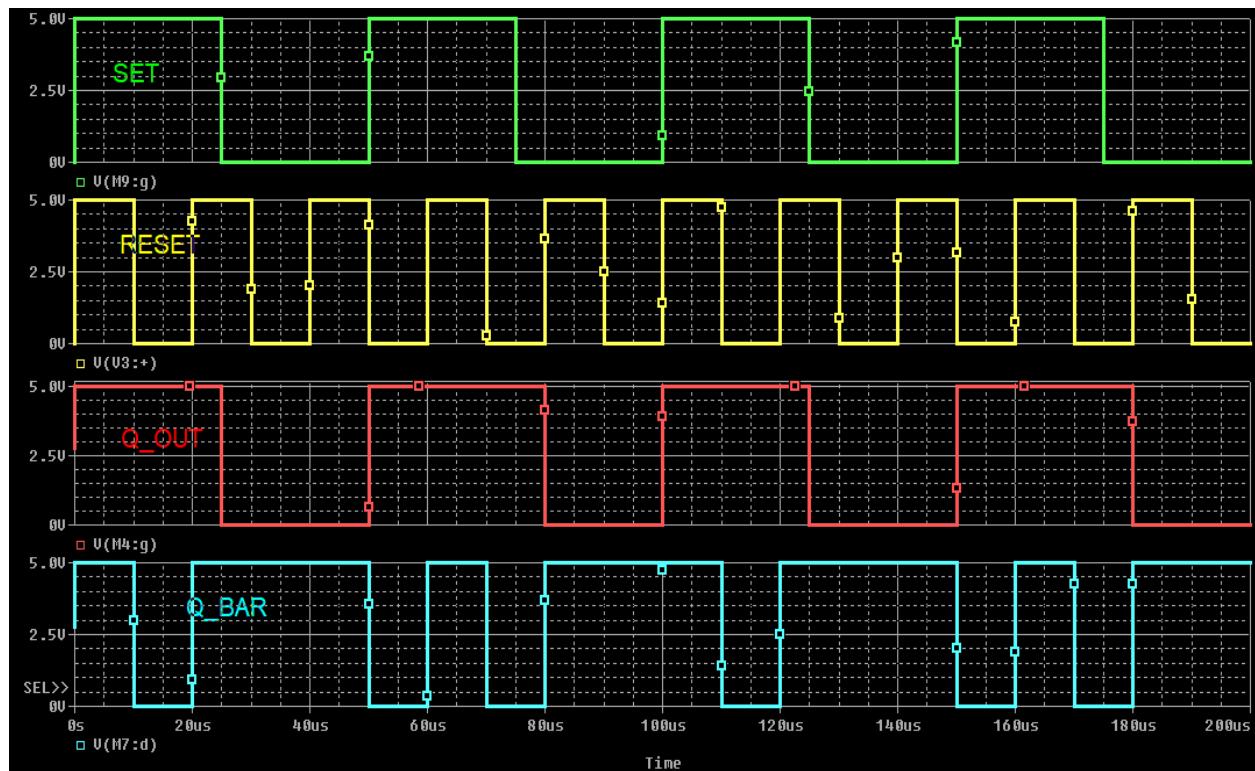
**Figure 2: D-Latch PSPICE waveform with voltage respect to time**

Optimal performance in retaining data is critical for the CMOS NAND-based S-R latch, where the sizing of transistors plays a pivotal role. In Figure 3, the PSPICE circuit features transistors sized accordingly to facilitate result comparison with the CD4007 CMOS inverter chip. The SR latch incorporates two pulse generators, each applied to the S and R inputs. These generators produce a voltage that rises from 0 V to 5 V, with S input operating at a frequency of 20 kHz and R input at 50 kHz.

In the waveform simulations of figure 4, the CMOS S-R latch performed as intended. When Set (S) goes to a logical high, the latch is in the “set” state and stores a binary 1. In this state, the Q output becomes high and Q-bar goes low. On the other hand, when the reset (R) input is activated or set to high, the latch is in the “reset” state and stores a binary 0. Q is now low and Q-bar becomes high.



**Figure 3: SR\_NAND PSPICE circuit with VDD=5V, VPULSE=5V @50kHz (1/0.02ms), VCLOCK=5V @20Khz (1/0.05ms)**



**Figure 4: S-R NAND PSPICE waveform with VDD=5V, VPULSE=5V @50kHz (1/0.02ms), VCLOCK=5V @20Khz (1/0.05ms)**

## **Applications:**

The CMOS D-Latch is an essential component in digital electronics, below are some of its applications.

1. Registers. In microprocessors D-Latches are key components in constructing registers within CPUs. Their job is to hold data while it is being processed.
2. Data Synchronization. They are used to synchronize data in digital systems, ensuring that data transfers occur in a controlled manner, aligned with the system's clock or control signals. They do this by delaying data transfer for a certain period of time.
3. Clock Domain Crossing. D-Latches allow for safe data transfer between different clock domains in systems with multiple timing sources.

The NAND-based S-R Latch is also a fundamental component in digital electronics. Some of its applications include

1. Debouncing Hardware. Latches are employed in debouncing circuits to stabilize noisy signals from mechanical switches. By filtering out unintended switch transitions latches ensure a clean and stable digital signal.
2. Flip-Flop Construction. Latches serve as a building block for more complex flip-flop configurations like the T, D, and JK flip-flops. These are necessary for creating memory elements for larger storage units like registers and memory arrays.
3. State Machines. In digital systems, the S-R latch is used to store the state of a system. Its ability to hold a state makes it vital in designing state machines for various control and logic applications.

## **Conclusion:**

In our experiment, we constructed and tested both D-Latch and NAND-based S-R Latch circuits in simulation and physically as well. The simulation phase employed MbreakN and MbreakP transistors from the PSPICE library, aiming to replicate the truth tables of each latch through CMOS technology. As demonstrated in figures 2 and 4, these simulations achieved successful outcomes. During the experimental setup, we encountered similar challenges as the previous lab when implementing the transmission gates necessary for the D-Latch. In contrast, the CMOS NAND gates used in the S-R Latch performed effectively, showing the expected behavior. This lab provided valuable insights into the functionality of various latches and their realization using CMOS technology.

California State University, Northridge

Department of Electrical & Computer Engineering

# ECE 442L - Digital Electronics Lab

Fall 2023



## Lab 7

### 4x4 NOR ROM Array

November 7, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

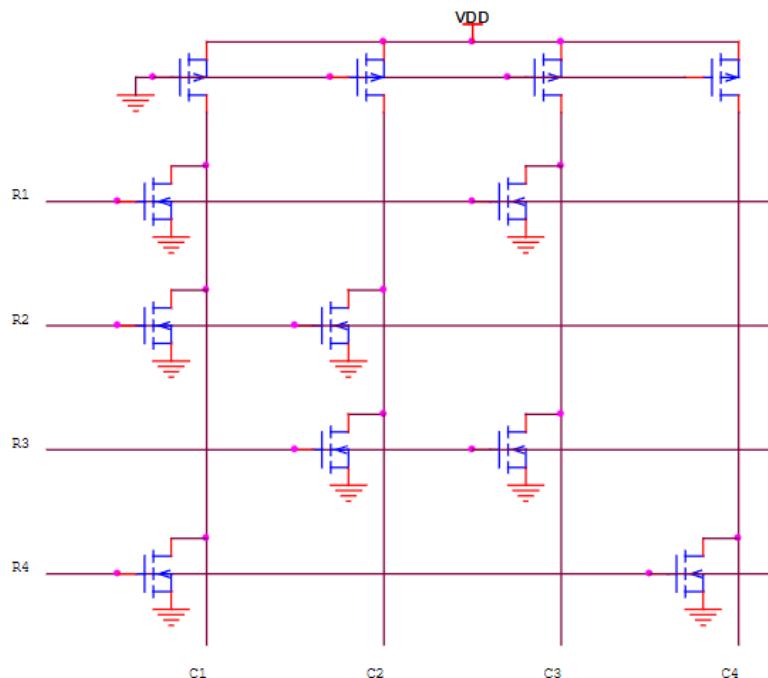
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## **Abstract:**

ROM is an important component in firmware storage. This experiment was aimed at designing, simulating and building a 4x4 NOR ROM Array. This was done by first simulating the ROM within PSPICE and getting an expected waveform from the circuit simulation. Then the physical circuit was built using multiple of the TI CD4007 chips. The build was successful and all expected outputs were seen from the two bit inputs. The objective was mostly met and observers gained understanding of CMOS latch construction

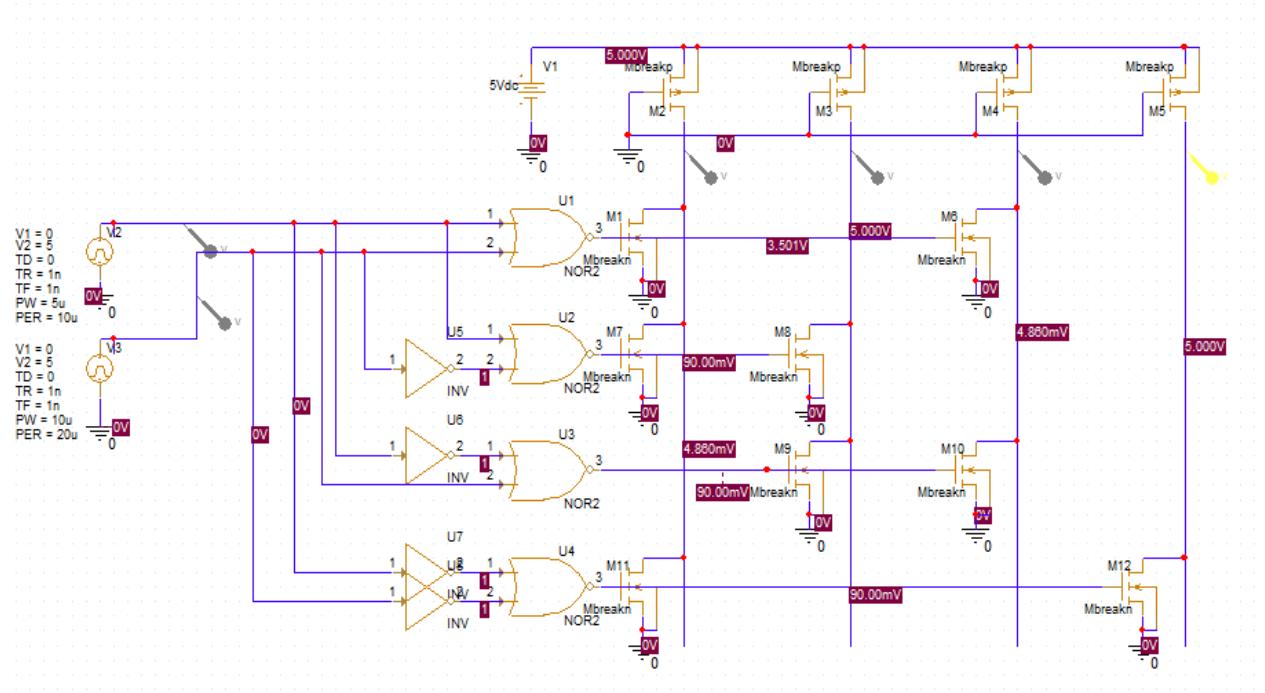
## **Theory:**

Complementary Metal-Oxide-Semiconductor (CMOS) Read-Only Memory (ROM) is an essential component in digital systems, serving as a non-volatile memory that stores fixed data, typically programmed during the manufacturing process. ROMs play a crucial role in a variety of applications, from storing firmware in microcontrollers to holding calibration data in sensor interfaces.

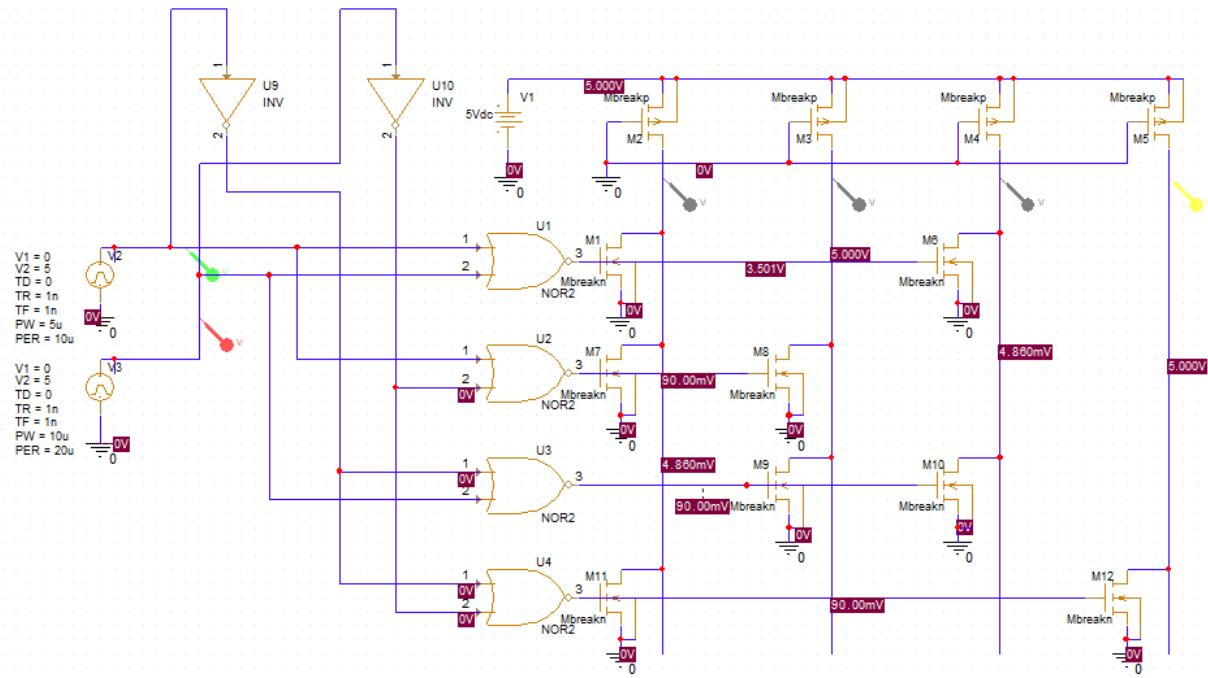


A CMOS ROM consists of an array of memory cells, each representing a binary digit (bit). These memory cells are arranged in rows and columns, forming an organized structure. The data stored in a ROM is fixed during the fabrication process and is not intended for modification during the normal operation of the device. During a read operation, the address decoder activates a specific row of memory cells based on the provided address. The data stored in the selected row is then read by the sense amplifiers and made available at the output for further processing.

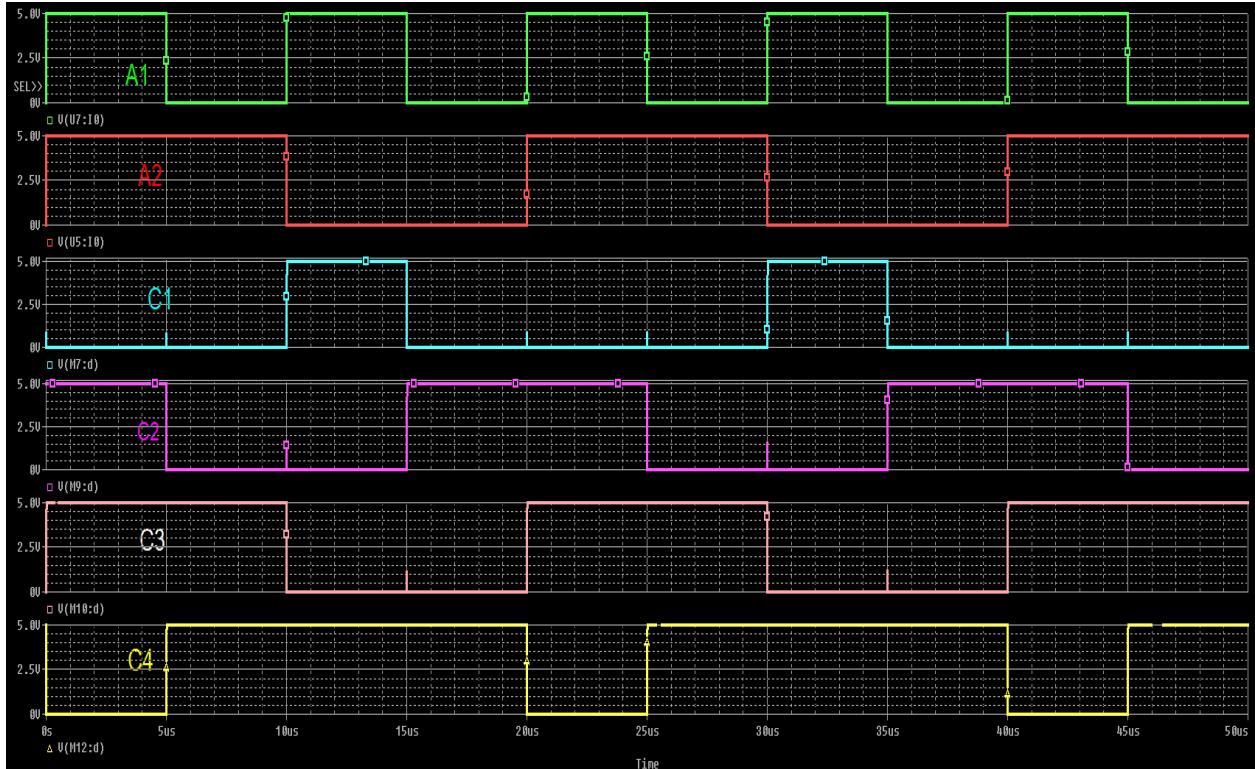
## Experimental Results:



1st version of PSPICE schematic design



Improved design of the 4 x 4 NOR ROM array circuit



PSPICE Waveform of a 4x4 NOR ROM Array

## **Applications:**

A NOR ROM array is a type of non volatile memory that stores data permanently. These are some of its applications.

1. Firmware storage. NOR ROM is widely used to store firmware which is software that initializes and manages hardware devices. Firmware is found in all kinds of devices like routers, TVs, cars, and other equipment.
2. Boot code storage. NOR ROM is ideal for storing boot code in computers and other electronic devices, as it allows for fast reading of the essential instructions needed to start up the system.
3. Embedded systems. Due to its fast read capabilities, NOR ROM is used in various embedded systems for storing control programs and operating system code. This is primarily in applications where the code does not need to be updated frequently, such as in calculators, simple electronic toys, and household appliances such as microwave ovens.

## **Conclusion:**

In summary, our project successfully demonstrated the functionality of a NOR-based ROM array, both in simulated and physical formats. The simulation conducted using PSPICE incorporated components like MbreakN, MbreakP, NOR, and INV to construct the ROM array. The successful replication of the expected outputs as evidenced in figure 3 indicated our accurate implementation of the ROM array in the simulation. For the physical experiment we did not use just the CD4007 chip as in previous labs, we added two different chips which were an AND gate chip and an inverting chip. This approach allowed us to dedicate the CD4007 chip solely for the ROM array. The experimental setup was successful, although we noted that the output representing a high did not quite reach the anticipated 5 volts of VDD. This project provided us with valuable insights into the workings of read-only memory and demonstrated how CMOS technology along with other components can be effectively utilized to implement such memory systems.

California State University, Northridge

Department of Electrical & Computer Engineering

## ECE 442L - Digital Electronics Lab

Fall 2023



### Lab 8

Seven Ring Voltage Control Oscillator

November 28, 2023

Instructor: Matthew Radmanesh

Written By: Peter Guzman, Cristian Robles, Zachary Gulshad

# Table of Contents

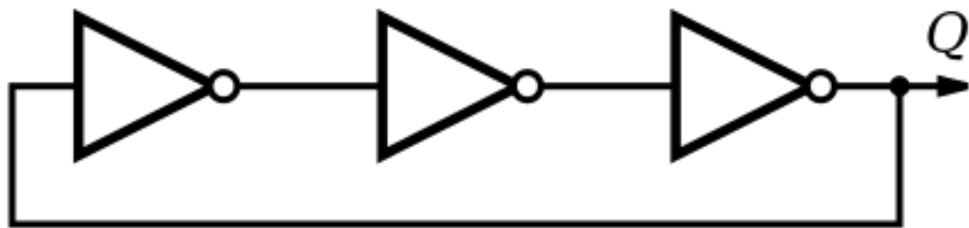
1. Abstract
2. Theory
3. Experimental Results
4. Applications
5. Conclusion

## **Abstract:**

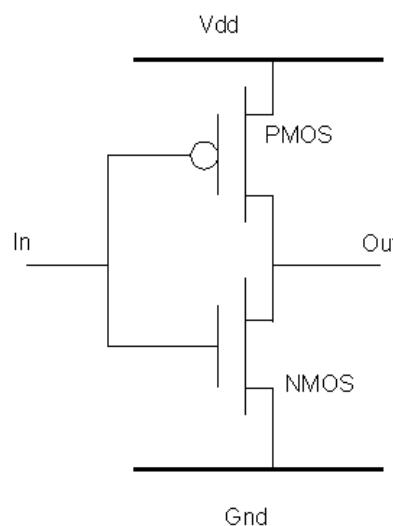
Digital clocks can be designed in a number of ways, one of which is chaining an odd number of inverters in a loop. This experiment aims at observing voltage controlled ring oscillator operation. This was done by designing a ring oscillator using various inverter stages, building such oscillators, and observing the frequencies when adjusting the voltage control, a varied frequency was observed as expected. The objective was met and a voltage controlled ring oscillator was built.

## **Theory:**

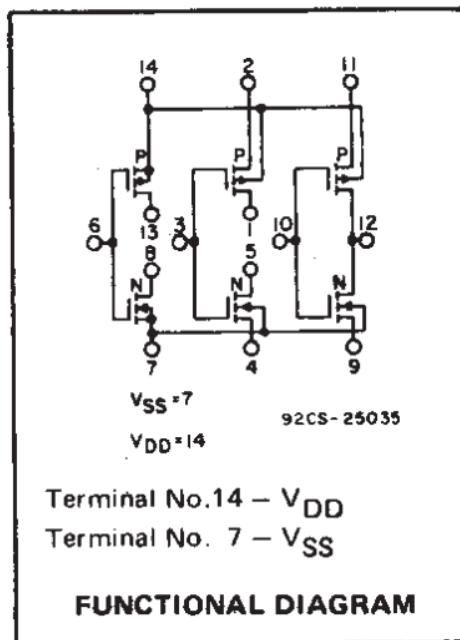
Ring oscillators behave as a clock switching between logic HIGH and LOW repeatedly. The oscillator is made up of inverters whose outputs are tied to other inverters inputs, this chain is made up of an odd number of links and the final inverter output is connected to the initial inverter input making the “ring”.



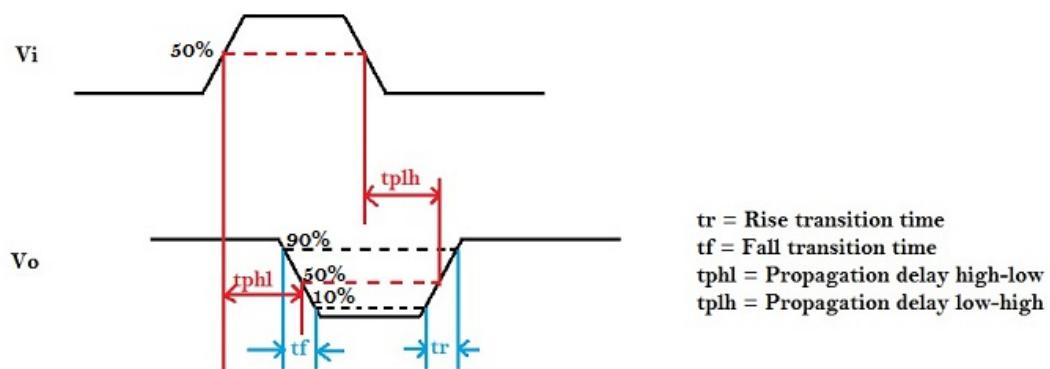
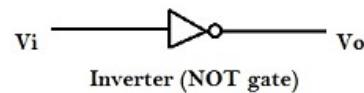
The invertors themselves are made up of CMOS technology transistors, the advantages of CMOS are simple structure, low power consumption, large noise tolerance and strong temperature stability. For the invertor a single NMOS and PMOS transistor connected in the way shown below.



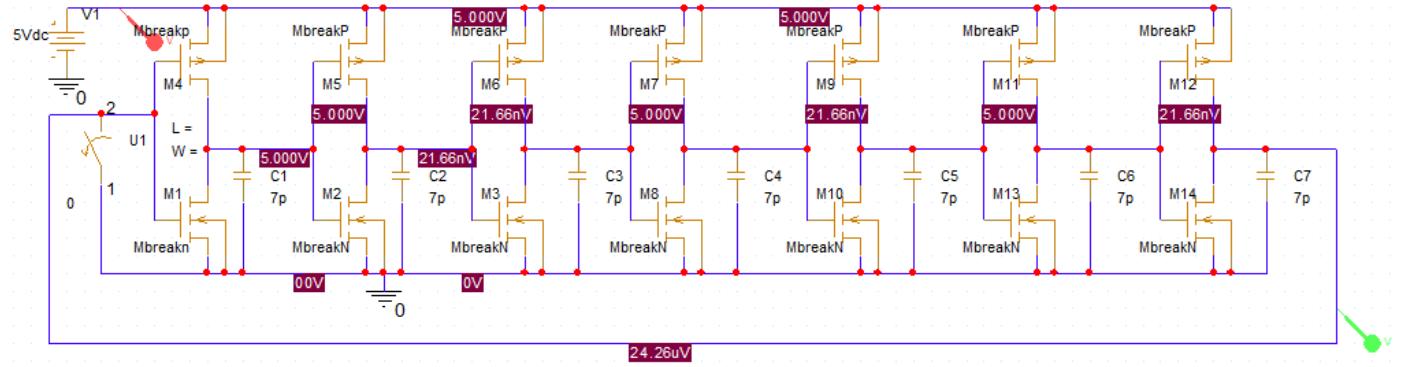
The chip used for the experiment is the Texas Instruments CD4007UB which is fitted with three pairs of transistors(each pair consisting of a single NMOS and PMOS).



The propagation delay is how long it takes a change in input to cause a change in output to appear, this value can be found by measuring from the half point of the input logic change to the half point of the output logic change as seen below as both t<sub>phl</sub> and t<sub>plh</sub>.



## Experimental Results:



PSPICE Schematic of Seven Ring Voltage Control Oscillator

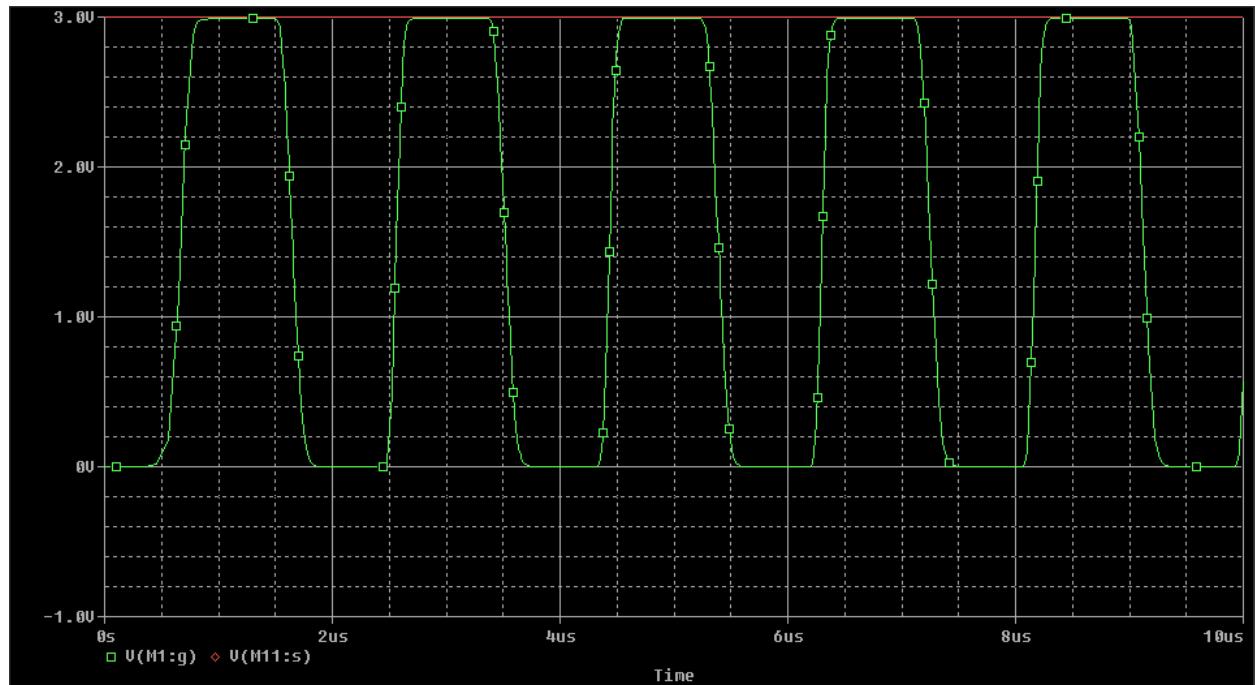
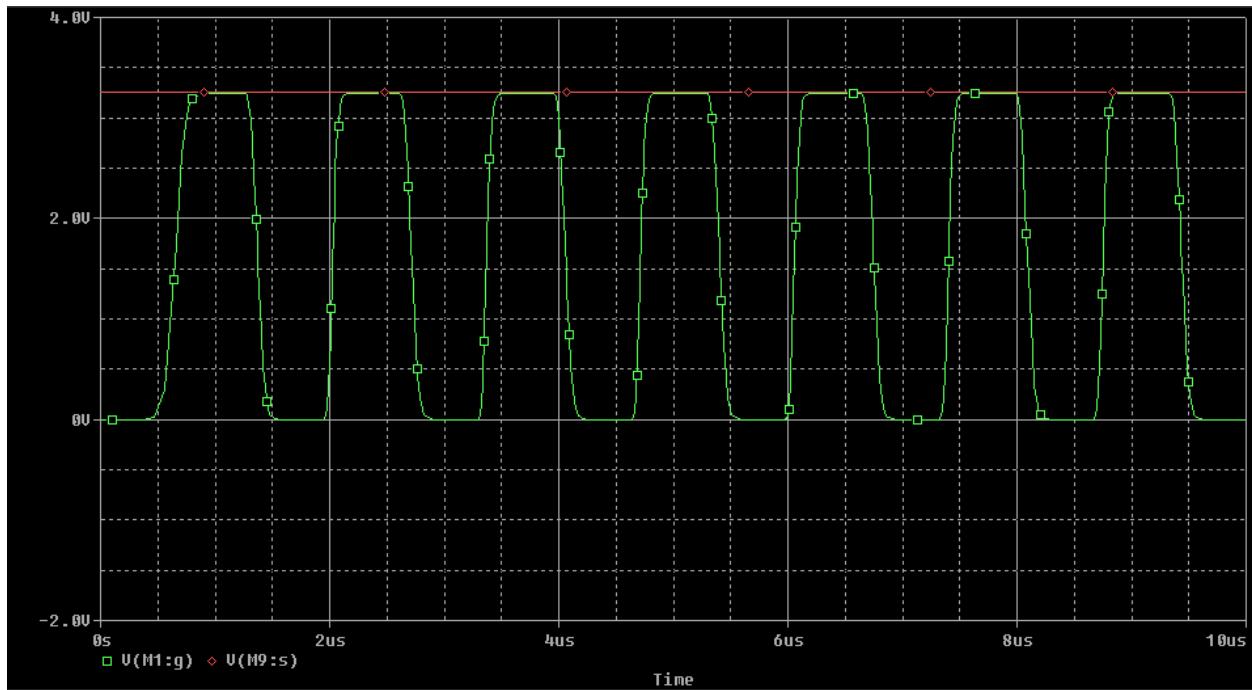
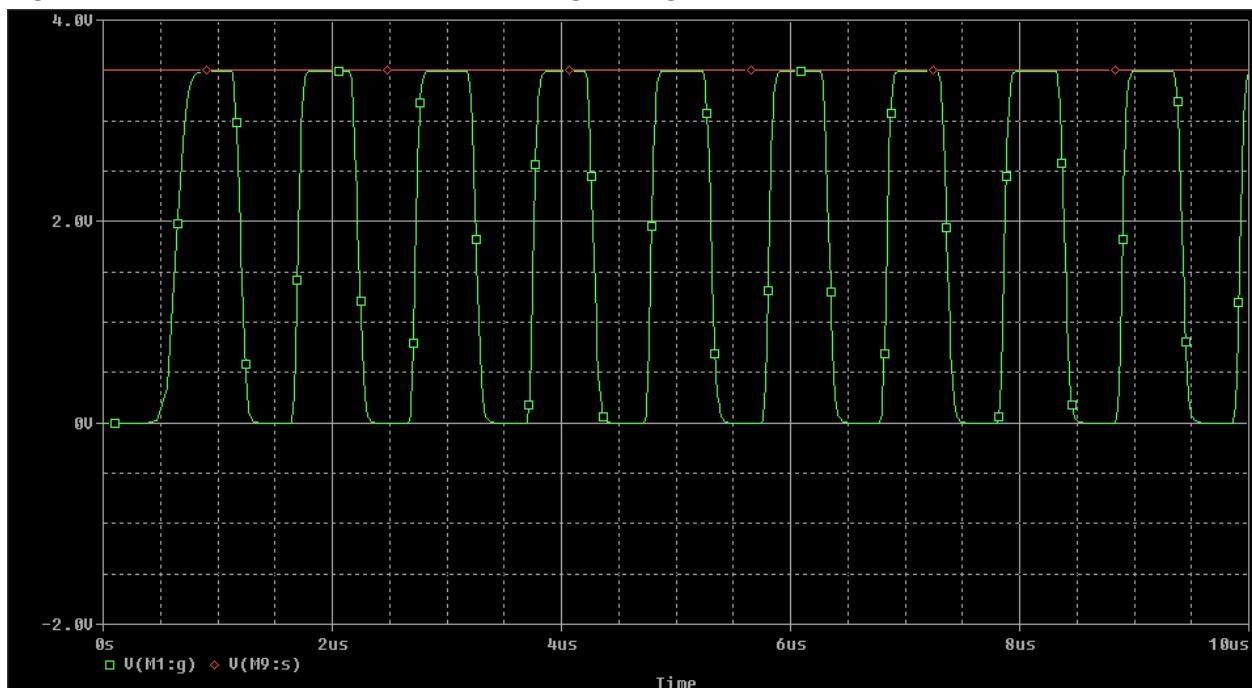


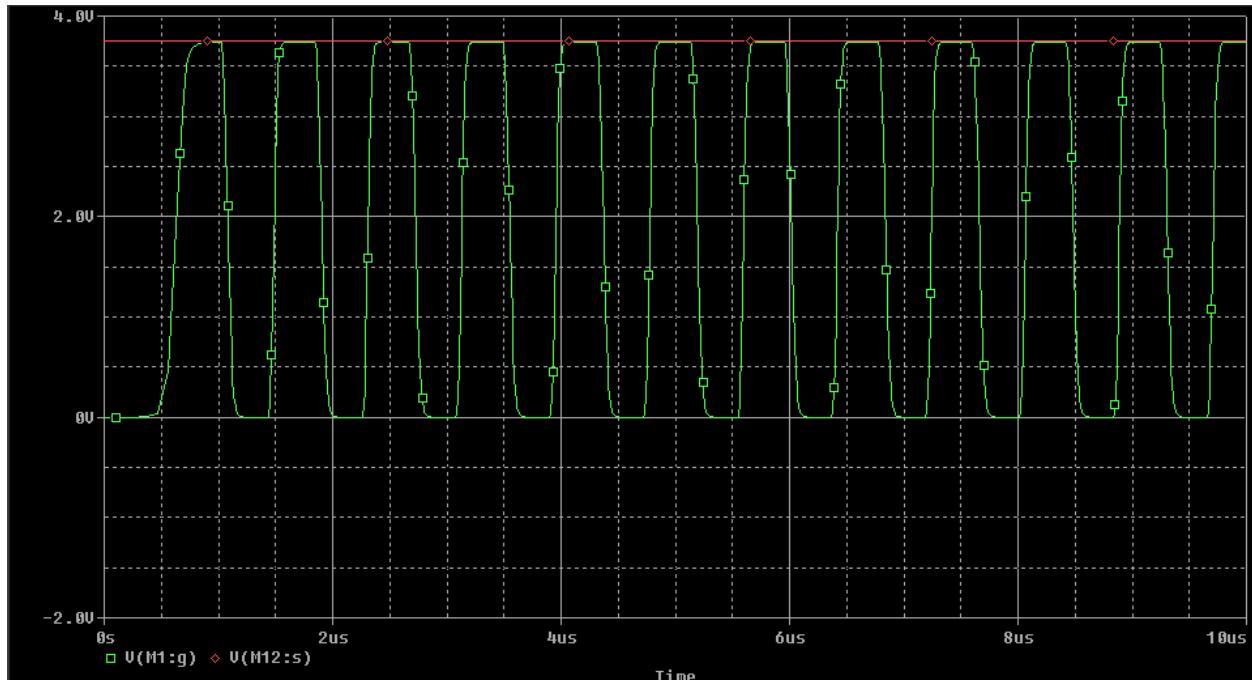
Figure 1 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 3V input



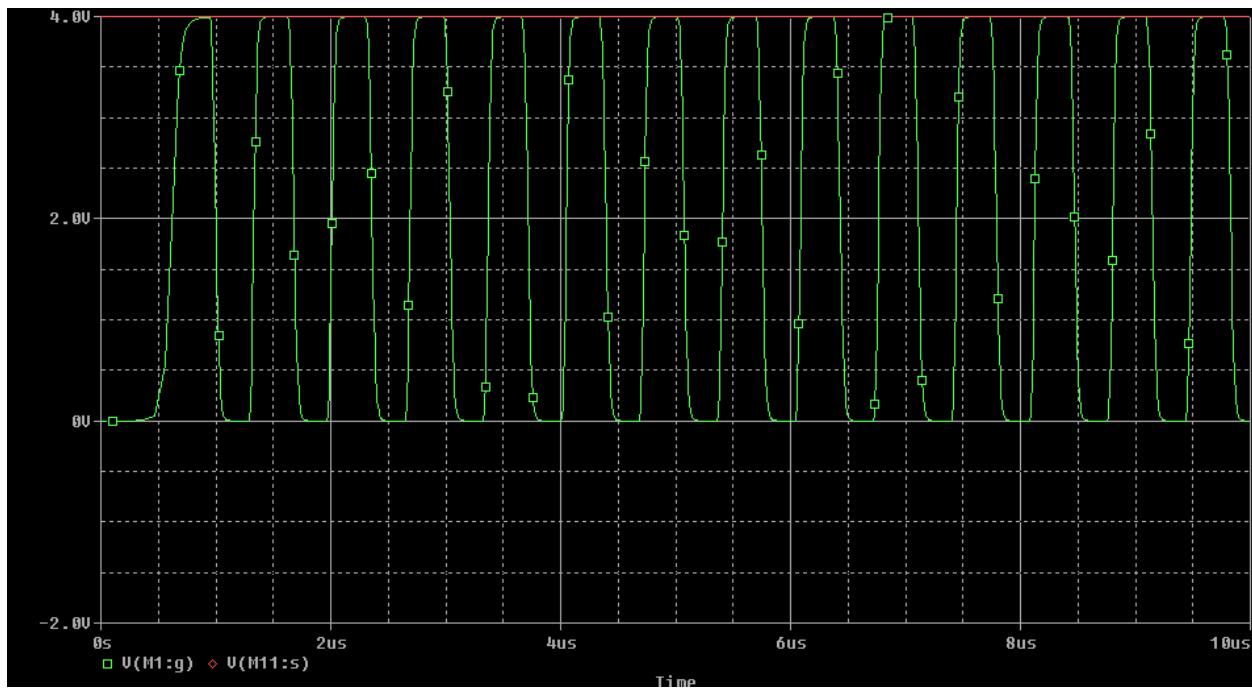
**Figure 2 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 3.25V input**



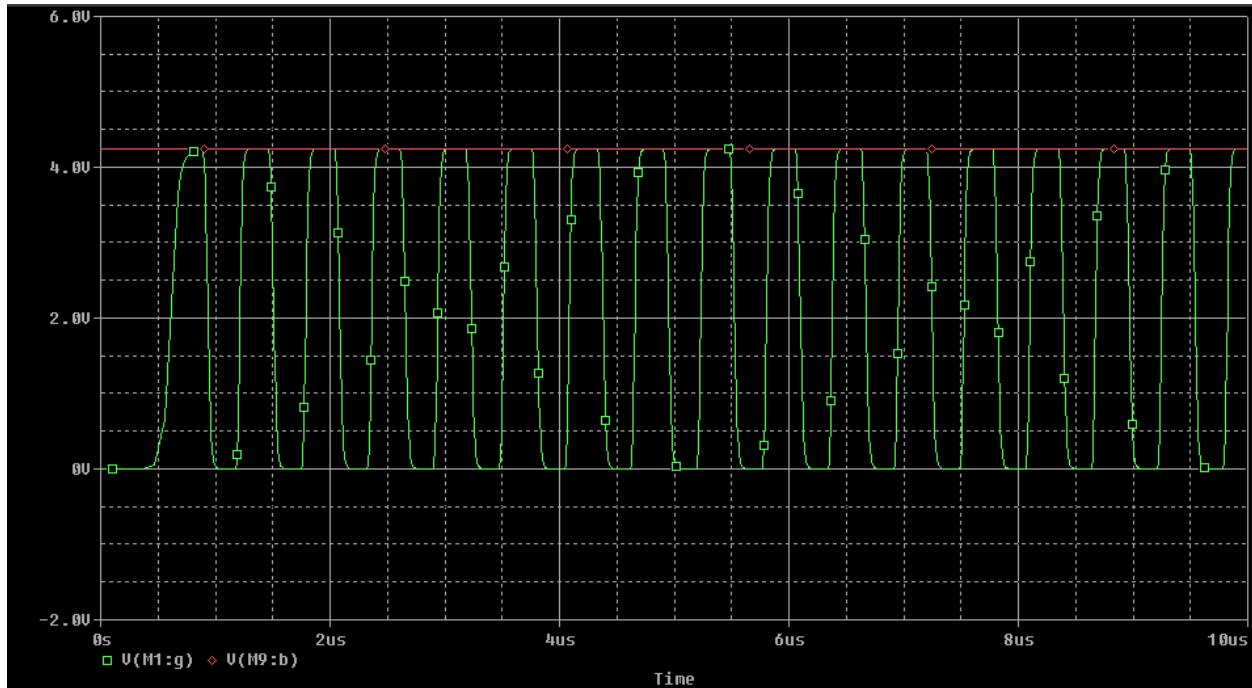
**Figure 3 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 3.5V input**



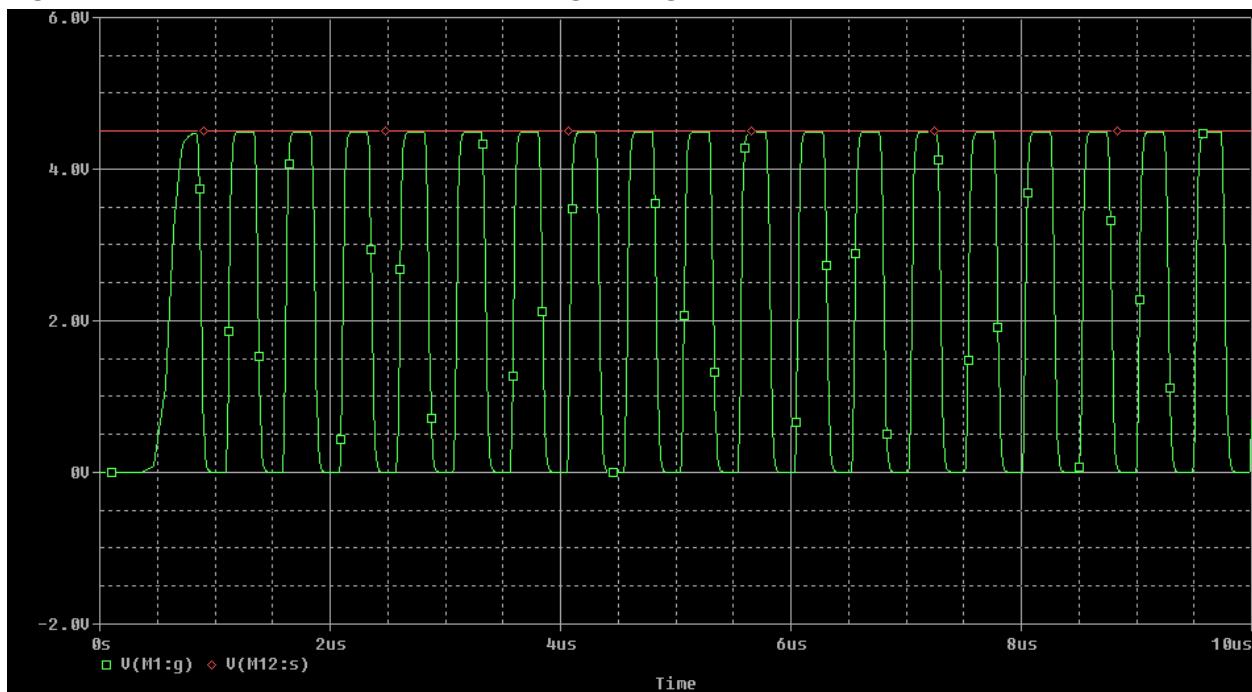
**Figure 4 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 3.75V input**



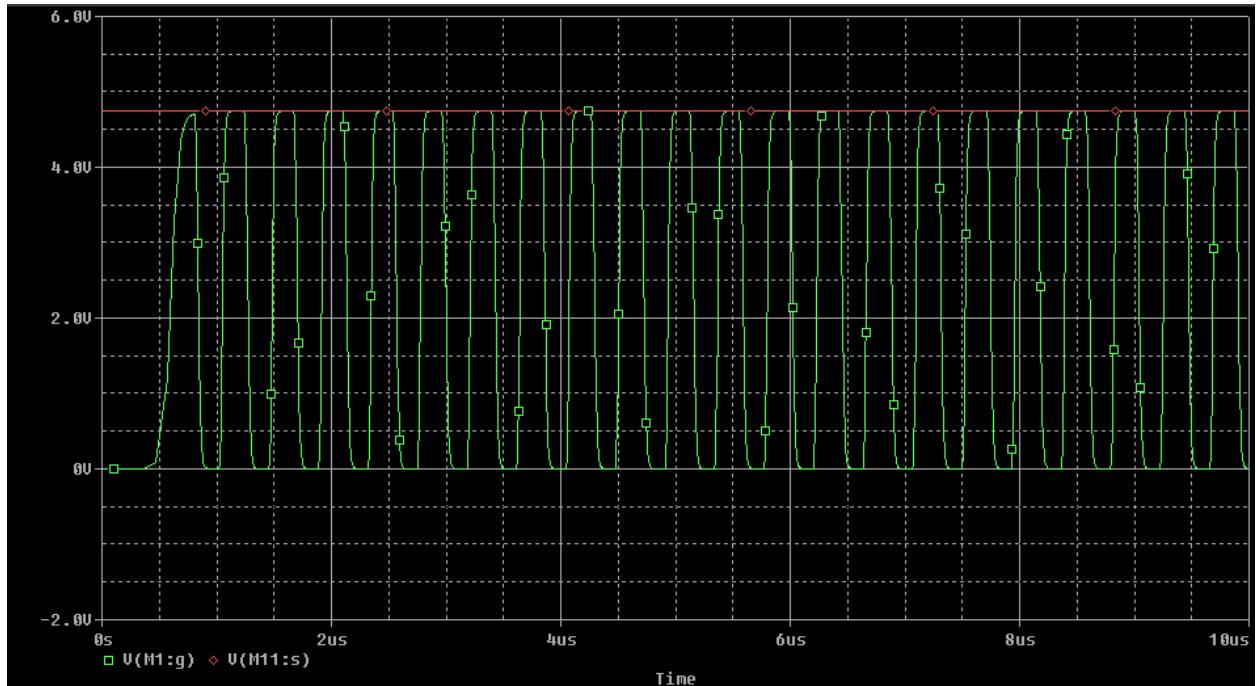
**Figure 5 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 4V input**



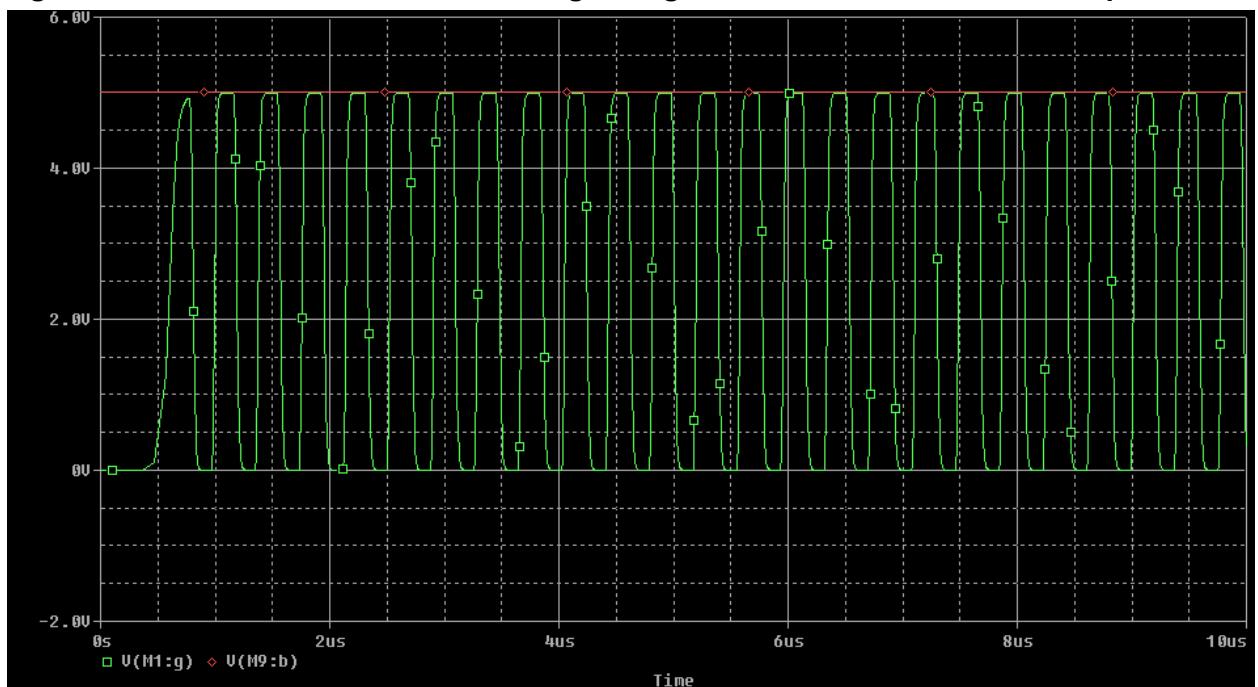
**Figure 5 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 4.25V input**



**Figure 6 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 4.5V input**



**Figure 7 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 4.75V input**



**Figure 8 PSPICE Waveform of Seven Ring Voltage Control Oscillator at 5V input**

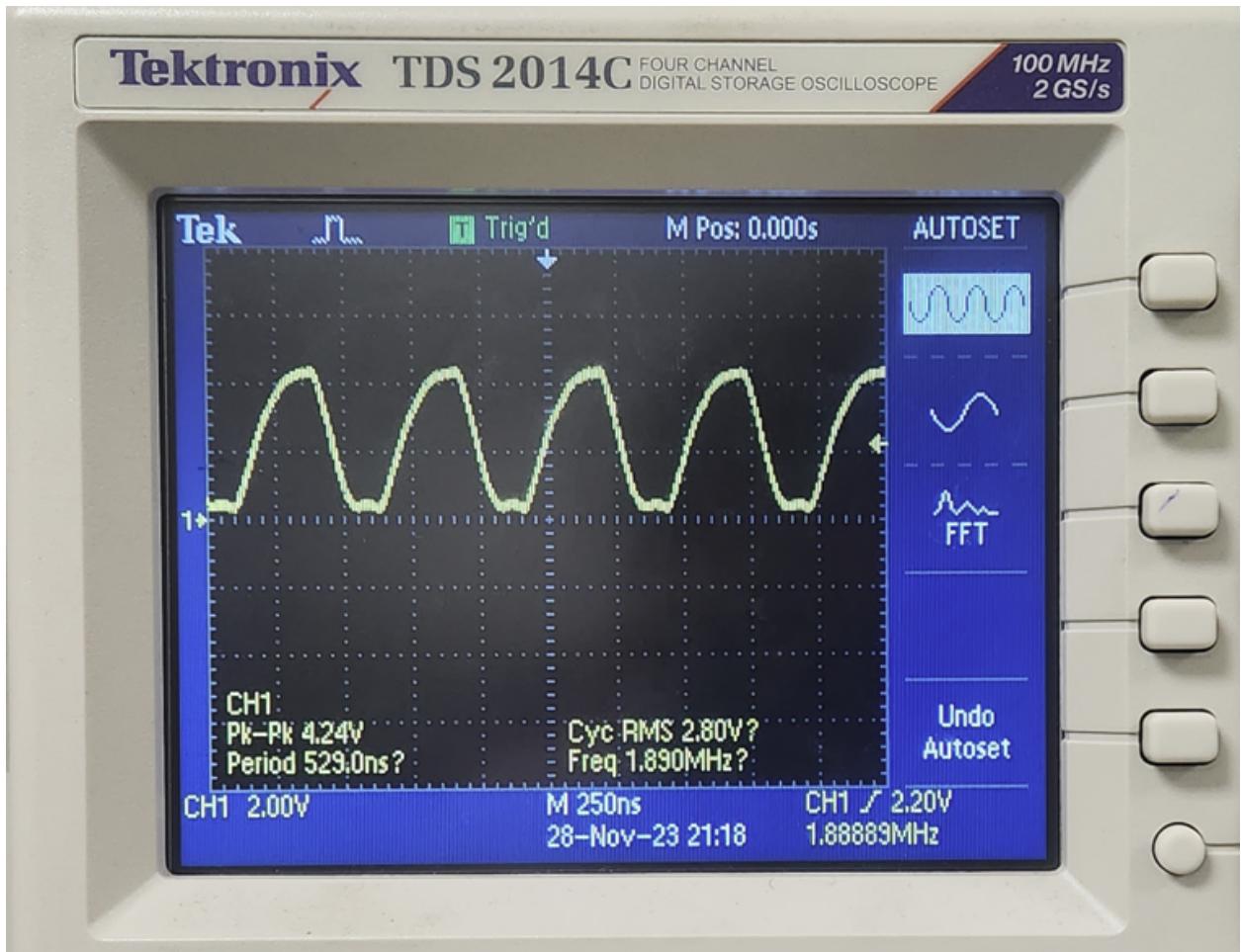
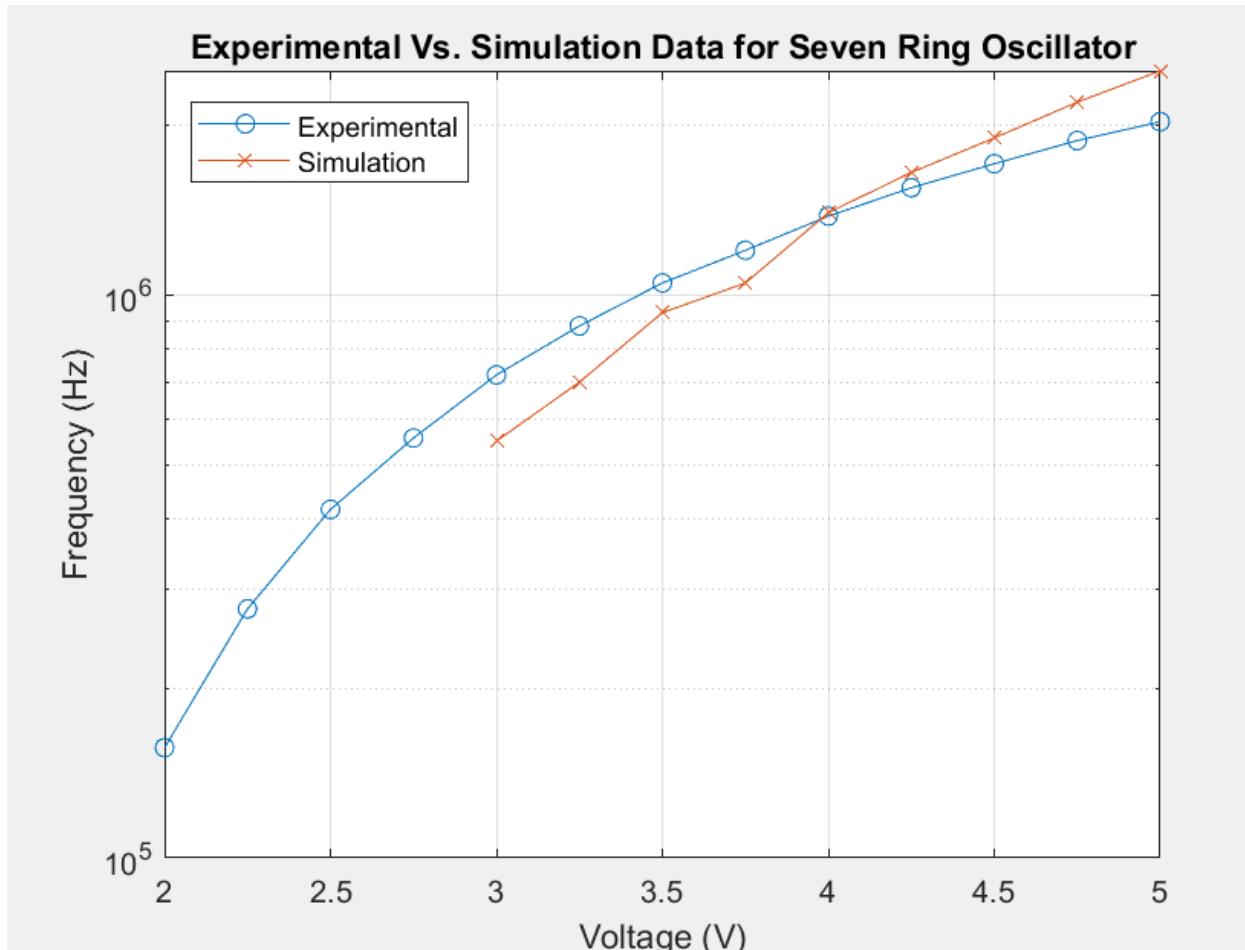


Figure 9 Experimental Seven-ring oscillator with 5V input



**Figure 10 Experimental Vs. Simulation Data for Seven-ring Oscillator**

### Applications:

A Seven ring oscillator uses a ring of seven oscillating stages. Its ability to generate a frequency modulated signal based on an input voltage gives it various use cases as seen below.

1. Phase Locked Loops. These circuits are for the synchronization of a variable oscillator with a fixed reference oscillator. Seven ring oscillators are a core component, they play a critical role in maintaining the lock between these oscillations in various devices.
2. Voltage and Temperature Sensors. Seven ring oscillators can act as sensors in integrated circuits, where changes in their oscillation frequency indicate variations in voltage or temperature.
3. Synthesizers. Seven ring oscillators are key components in synthesizers, used in music production and audio engineering. They generate various audio tones and frequencies, enabling the creation of a wide range of sounds and musical effects.

### Conclusion:

In conclusion, both the simulation and the physical construction of the seven ring oscillator circuit demonstrated successful operation. In the simulation phase, we used MbreakN and

MbreakP transistors within PSPICE to construct CMOS inverters. These inverters were then arranged in a series and configured into a ring formation, enabling us to apply a voltage and accurately measure the oscillation frequency. For the experimental part of the lab, we employed the CD4007 chip to create a similar ring of seven inverters. Initially, we encountered a wiring issue that prevented the oscillator from producing the correct output. However, after this error was resolved, the oscillator functioned as expected. The experimental results, as depicted in Figure 10, closely align with those from the simulation, confirming the consistency and reliability of our approach. This lab effectively illustrated the application of CMOS technology in generating oscillations, a principle that can be applied across various scenarios in electronics.