The KD1S Modulator:

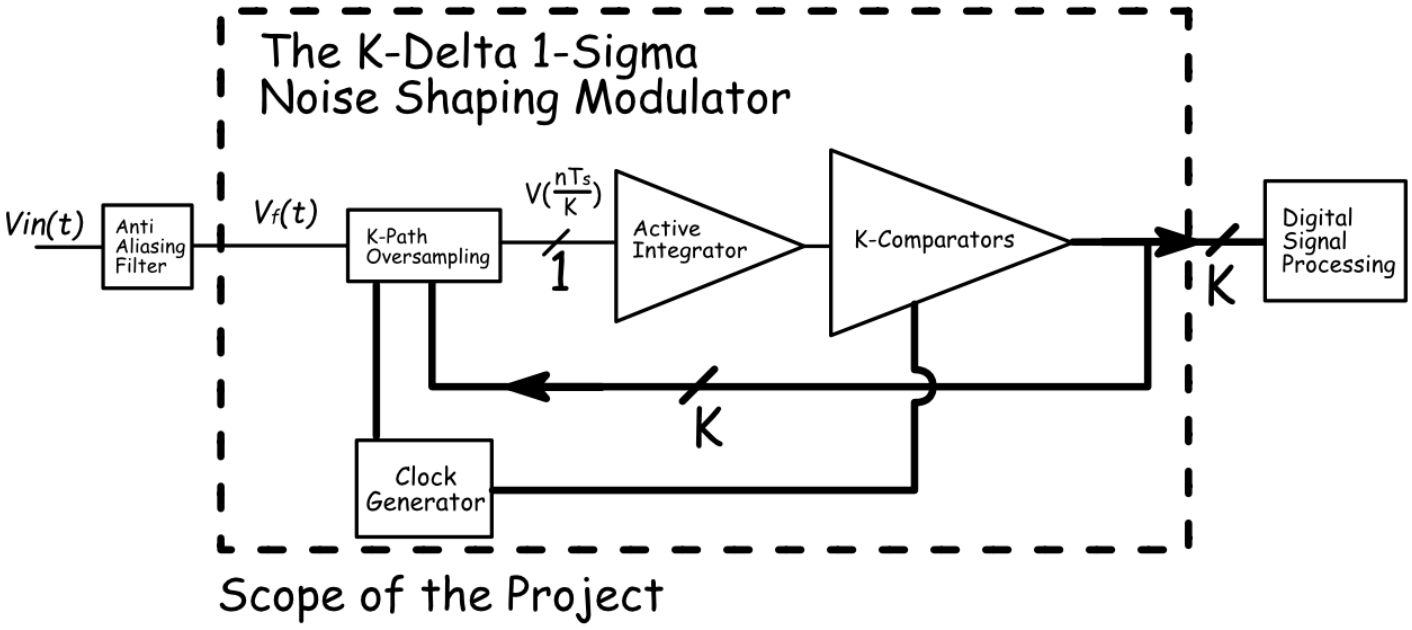
By David Santiago
ECG 781 – Mixed Signal Circuit Design
Fall 2020

Abstract

The purpose of this project is to demonstrate the properties of a noise-shaping modulator with different topologies. For this report, we will be trying to use theoretical hand-calculated models to predict whether a topology will work in simulations and in practice.

In signal processing, one of the tools used is the Z-Transform and Z-Domain, which is a useful tool for demonstrating whether a system will be stable by looking at the location of poles. Another tool that is most often used in signal processing is the Fourier Transform, or a spectrum, which is used to analyze signals and viewing the spectral contents such as information and noise. These tools will be useful for designing analog/digital filters so that certain information may be obtained while getting rid of noise or unwanted data.

A roadmap of what is needed is shown in the figure below:



The input signal to the circuit must have limitations on the frequency that may be inputted. The Anti-Aliasing Filter (AAF) is placed at the beginning of the circuit so that the signal does not create unwanted images in the desired spectrum.

The K-Path block represents the input signal using K-Path Sample-and-Hold techniques to overall increase the sampling rate by a multiple of K.

Note that the general rule for Sample-and-Hold Analog-to-Digital Conversions (S/H ADCs) is that:

$$f \ll f_s$$

This is due to the S/H having the properties of a natural Low-Pass Sinc Filter, where at the output of each individual K-Path:

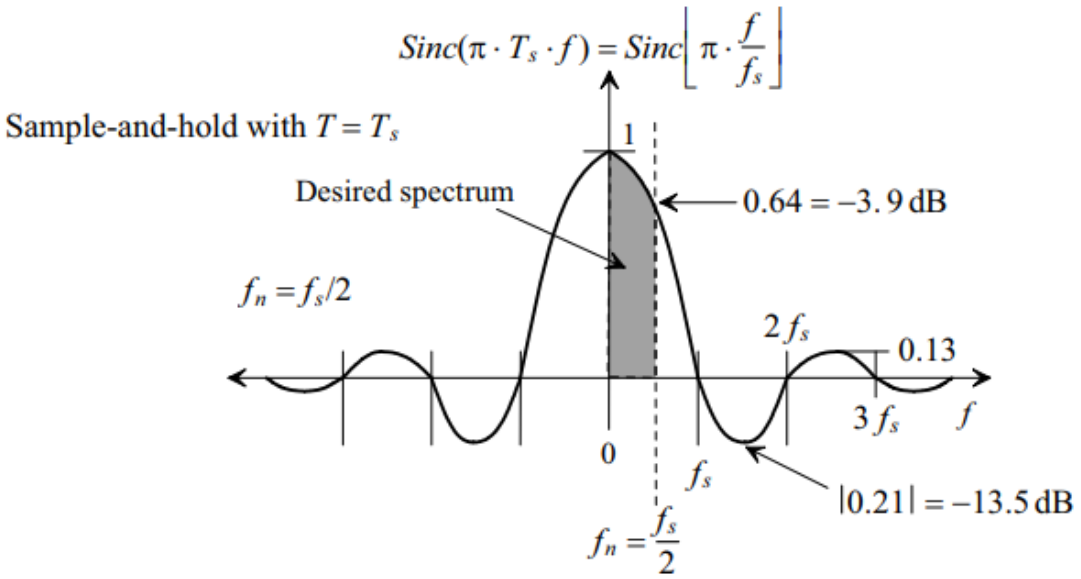


Figure 2.17 The frequency response of a S/H.

Where the Nyquist Frequency of a single path is:

$$f_n = \frac{f_s}{2}$$

Even though K-Path sampling is used to oversample the input signal, the input signal is still individually inputted through a single path that is sampled at the sampling frequency, f_s.

However, the benefits of using a summation circuit (Active Integrator) is that although that a path may not be fast at sampling, the other paths will sample the input signal at different phases and adding all of the results into one path, effectively capturing more data and increasing the bandwidth of the general S/H ADC filter!

The new limitation with K-Path sampling and having 1 summation is:

$$f \ll f_{s,new} = K \cdot f_s$$

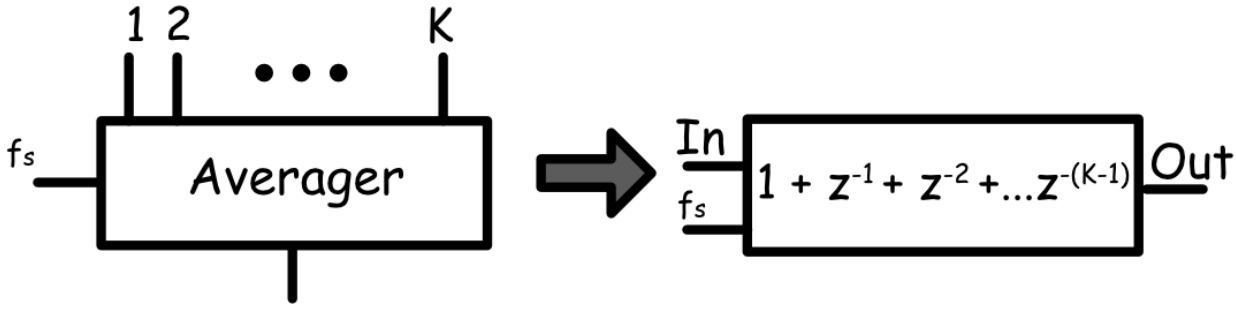
And the new Nyquist Frequency for our desired signal is:

$$f_N = \frac{f_{s,new}}{2} = \frac{K \cdot f_s}{2}$$

This limit can be used as a design parameter for the Anti-Aliasing Filter, so that any frequency above this bandwidth may be filtered out so that no aliasing will occur in our circuit.

Note, although our Nyquist frequency may be high, however, our input signal will be limited due to the bandwidth of the Digital filter.

Recall a simple averaging filter of K paths:

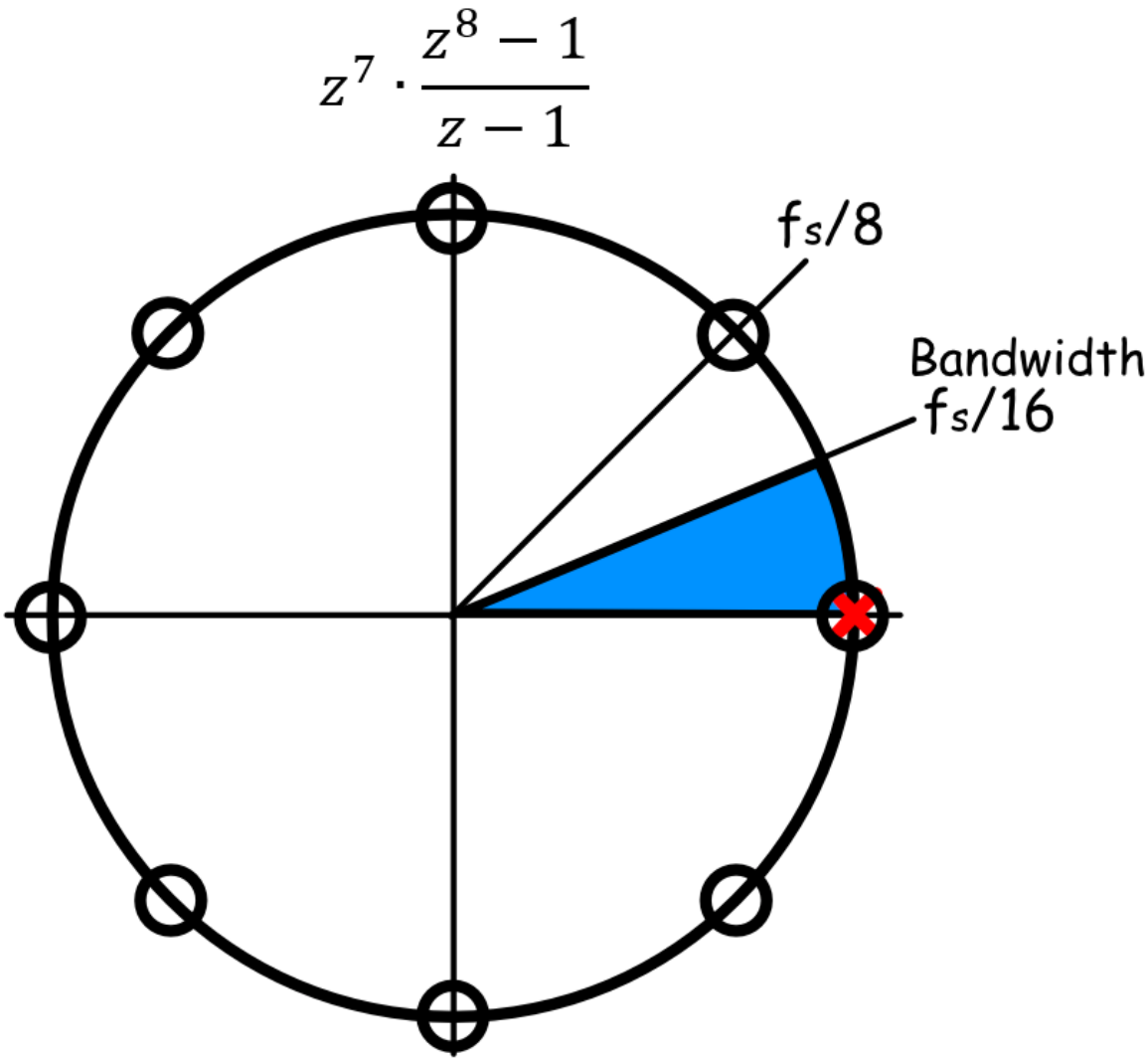


All of our outputs will be tied into one register that will send bits to MATLAB at a clock frequency of f_s. Due to recombining the signals, this effectively acts as an averaging filter of K values.

Therefore, analyzing the averaging filter a pole-zero diagram:

$$1 + z^{-1} + \cdots z^{-(K-1)} = \frac{1 - z^{-K}}{1 - z^{-1}} = \frac{z^K - 1}{z - 1} \cdot z^{K-1}$$

For K = 8:



Due to this averaging effect, our overall input frequency into the Noise-Shaping Modulator should be:

$$f_{Bandwidth} = \frac{f_s}{2K}$$

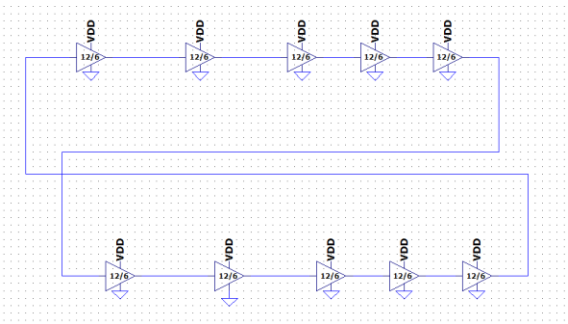
For this project, since the input signal will relatively be slower than the new sampling frequency, an Anti-Aliasing Filter will not be needed.

Before implementing the noise-shaping modulator, a few things must be designed.

- Clock Generation Circuit
- Differential Amplifier
- Clocked Comparator

Clock Generator

For this design, we will be looking at a simple ring oscillator.

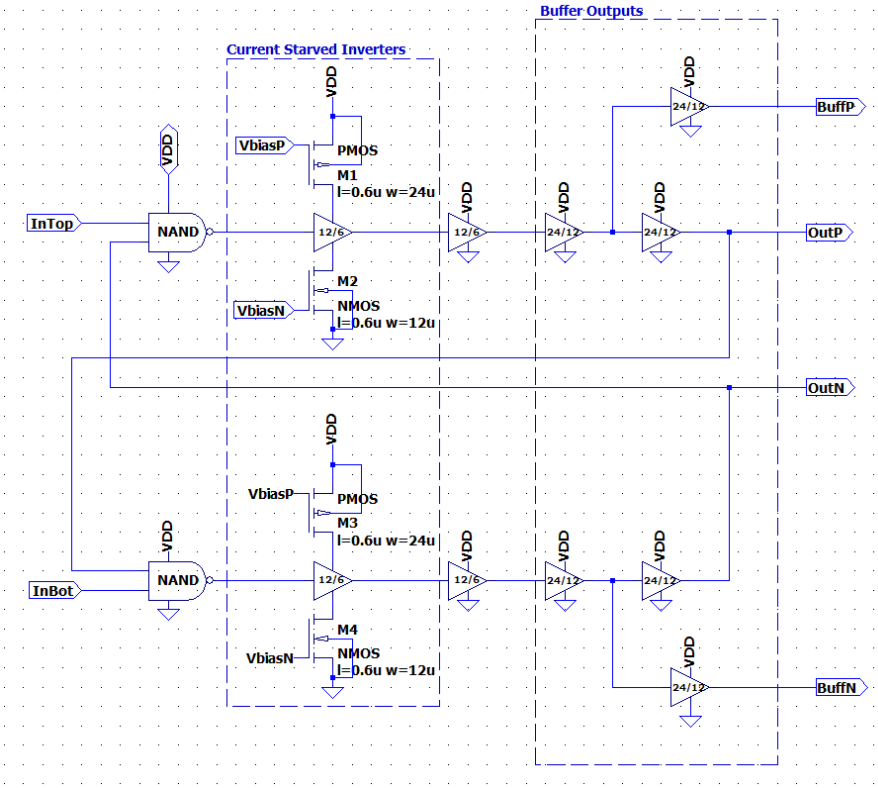


The positive of this topology is that it is simple and small. Also in terms of clock jitter, this circuit is very prone to clock jitter and if looking at a single inverter, the jitter that goes into an inverter is the same that comes out (it is consistent).

The negative is that this design is hard to tap into as any loading will cause differences in pulses.

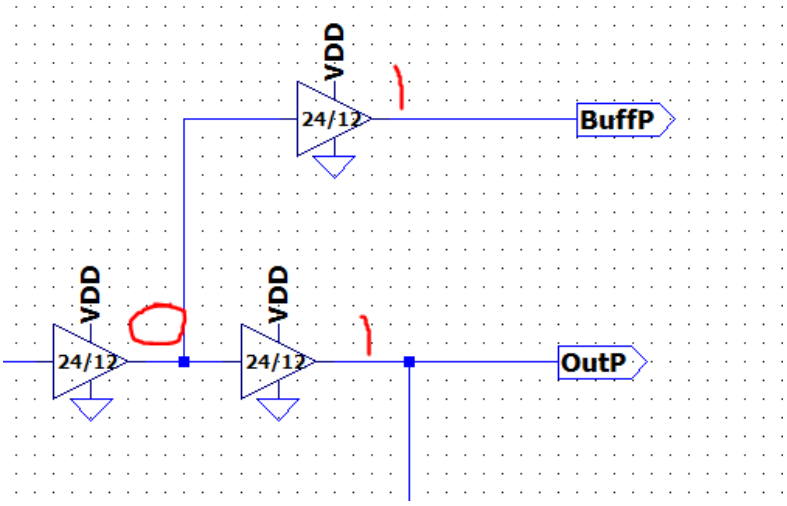
Note: All transistors will use a minimum length of L = 0.6µm, unless specified to have a different length.

A new topology is implemented (based on the topology in the Mixed Signal Design book, Fig 9.23):



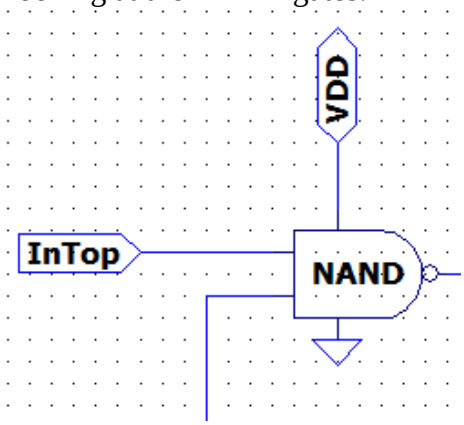
With this topology, a buffer is added in the ring oscillator, which makes it possible to tap into the circuit and carry out the signal. Also, a current starved inverter is placed into the circuit.
For layout purposes, 2 current starved inverters are placed so that each branch is ideally symmetrical.

Analyzing a node:



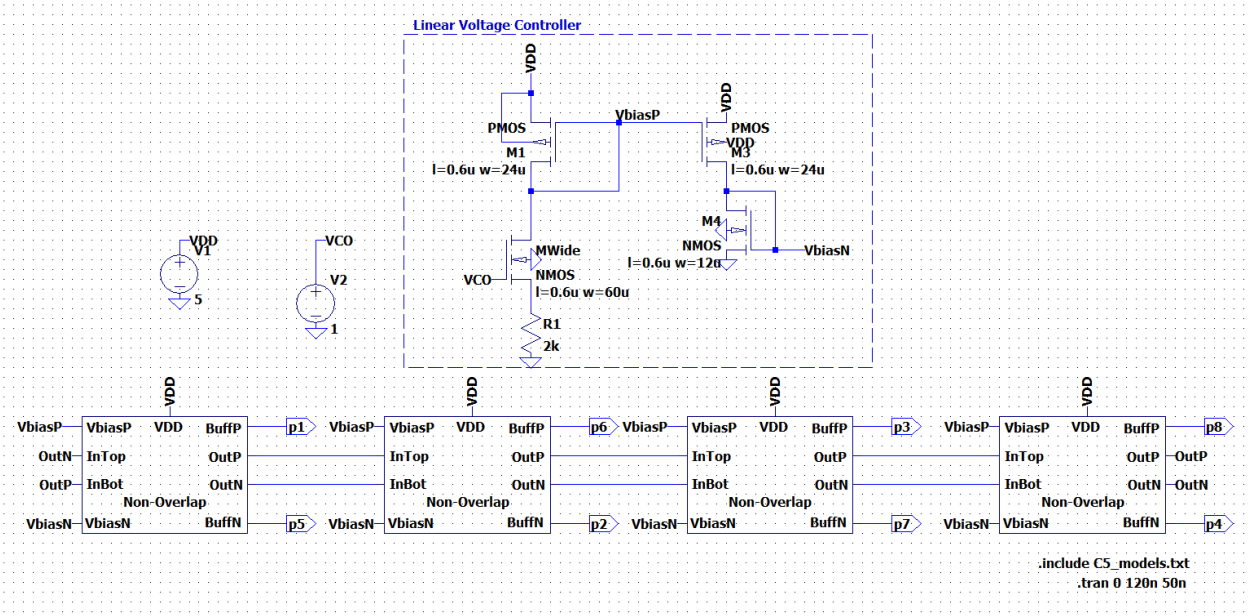
For clock jitter to be consistently similar, a 24/12 PMOS/NMOS width ratio is implemented. If there are any process changes or errors, with proper layout techniques, the clock jitter that comes out of the 2 inverters should be very identical.

Looking at the NAND gates:



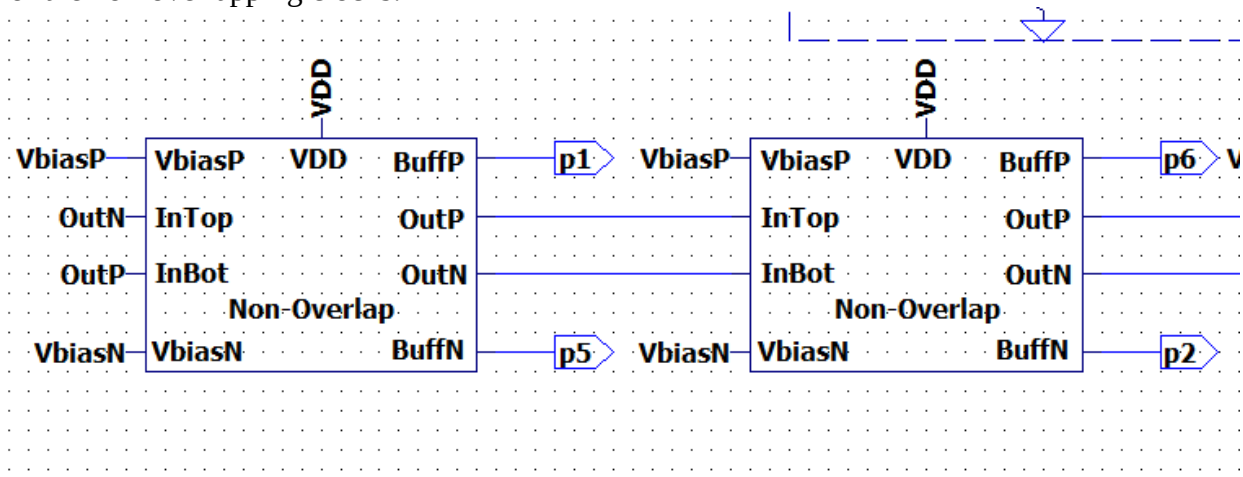
This is to be used for creating multiple stages to implement overlapping clocks.

Creating the Clock generation circuit:



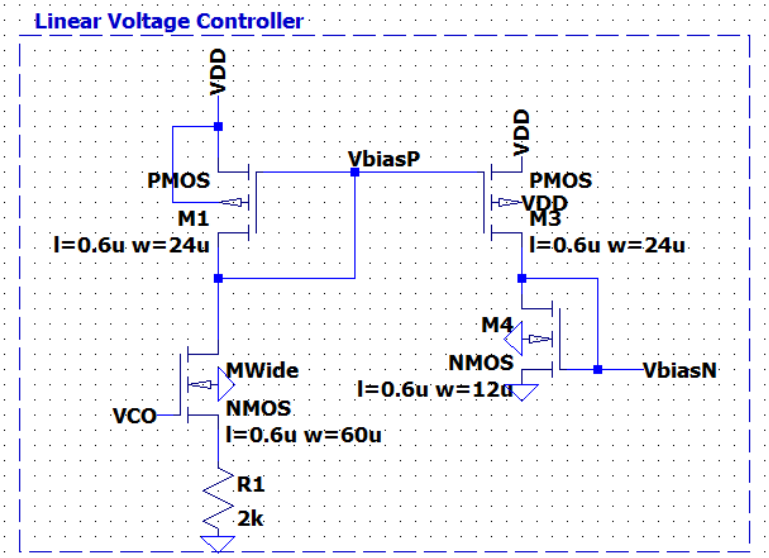
The clock generator consists of 4 non-overlapping stages, where the final stage is inputted back to the beginning and has outputs crossed over to create the correct oscillation.

For the non-overlapping clocks:



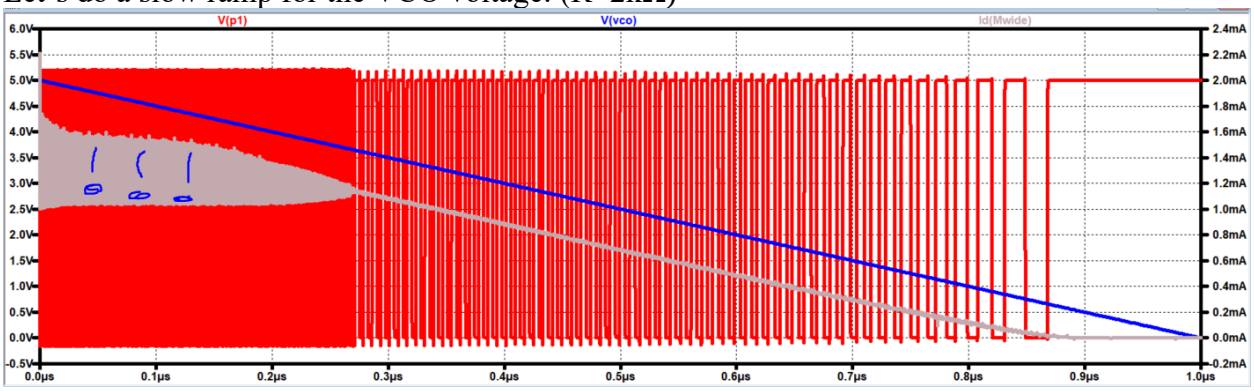
Outputs P1 and P5 are logic 1 and logic 0, respectively. Outputs P2 and P6 will also follow a similar logic, and be out of phase with respect to the first chain by about $1/4^{th}$ of a clock cycle.

Looking at the following circuit used for the current-starved inverters:



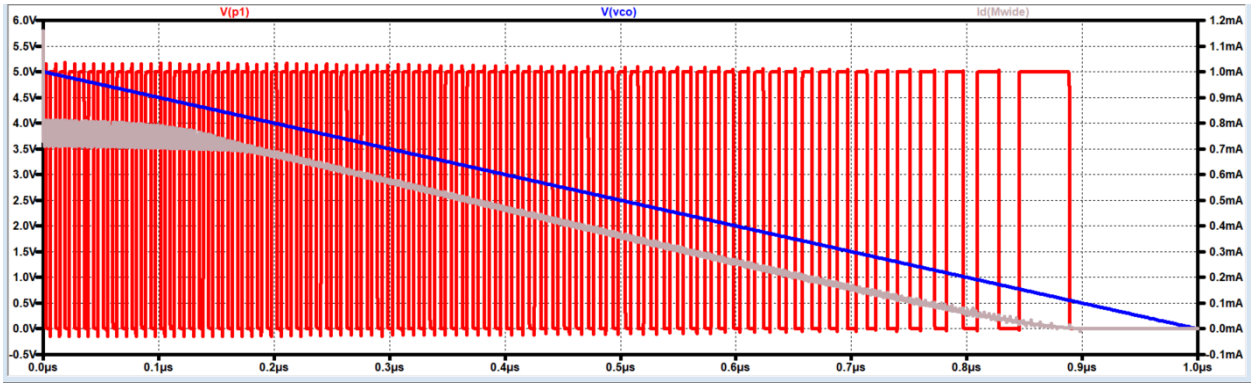
This is used so that we are able to linearize the frequency as a function of voltage. Note for this to work, a wide NMOS device is used, noted above as "MWide". This is so that the V_{GS} of the device is effectively $V_{TH,N}$. The current that is pulled from the PMOS M1 will be mirrored on PMOS M3. For currents to match in both NMOS and PMOS devices, a 24/12 ratio was chosen. This effectively doubles the current, and will help increase the frequency in the current starved inverters if needed. Note that the input voltage VCO does not have to be at its maximum, and can be set by the user if needed.

Let's do a slow ramp for the VCO voltage: (R=2kΩ)



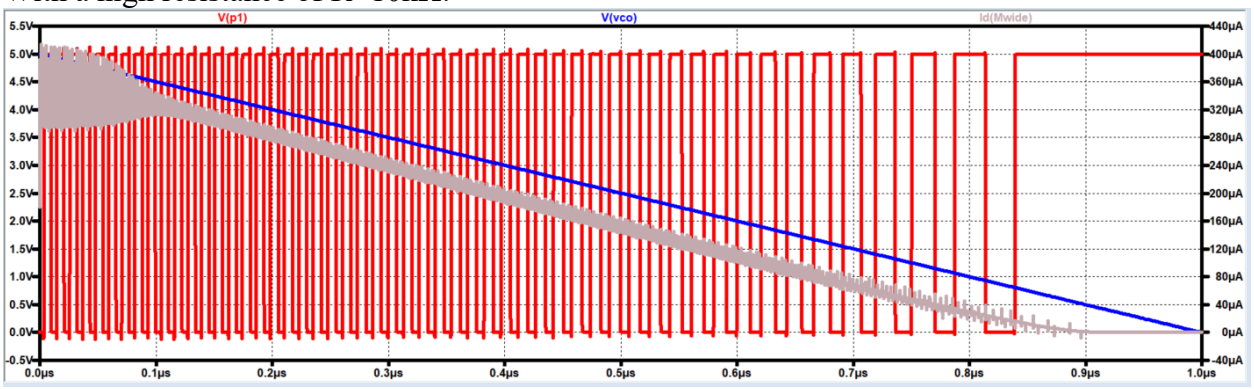
With a lower resistor, we see that we have an inconsistent current through the wide NMOS with high VCO voltages. This is not desired, as the user should be able to have some control over the frequency and have linear current.

Redoing the simulation with R=4kΩ:

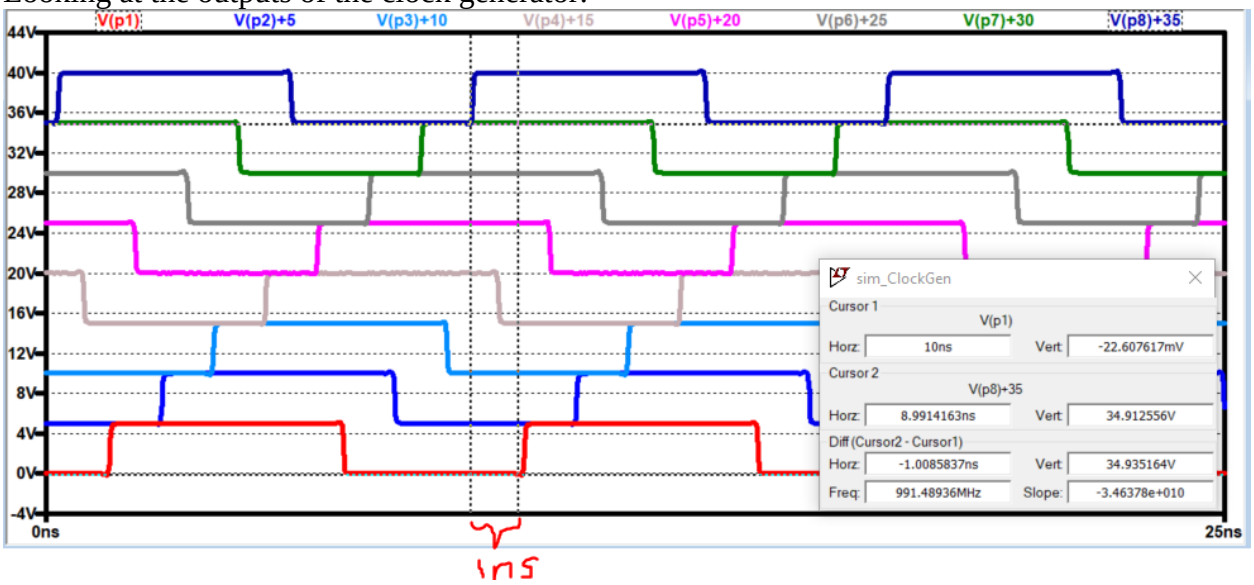


The current here is more linear and at high VCO voltages (VCO = VDD), we should have a much more stable frequency.

With a high resistance of R=10kΩ:

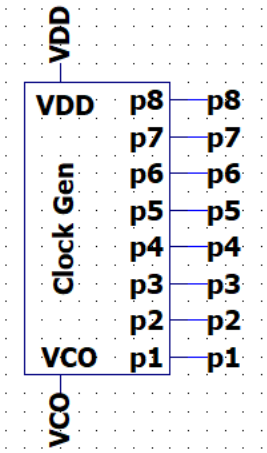


We see that our frequency has slowed down and also the current at high VCO voltages isn't consistent. Therefore, our resistor should be around 4kΩ for consistency.
Looking at the outputs of the clock generator:

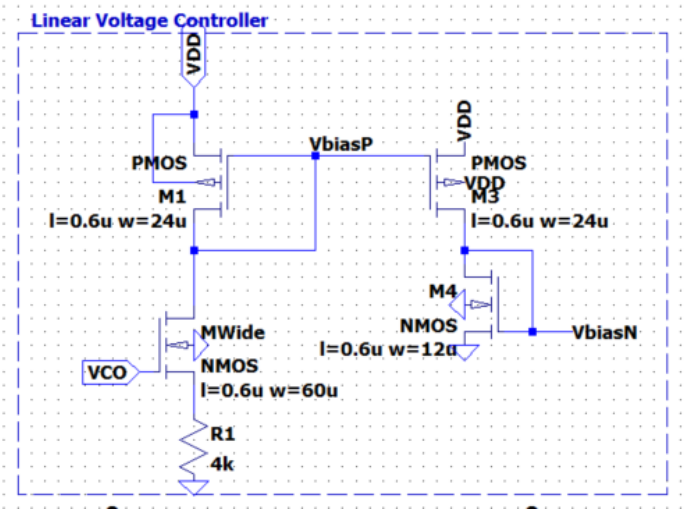


The time the clock is high is greater than for when the clock is low (duty cycle >50%). For K-Path sampling, this is ok as we will only focus on the rising edges of the clocks. Note that the duty cycle can be changed by adding delay in the non-overlapping stages. Also note that inverting all the signals results with a duty cycle < 50%, if desired.

Our final symbol:



Final Linear Voltage Controller:



Characteristics of the Clock Generator:

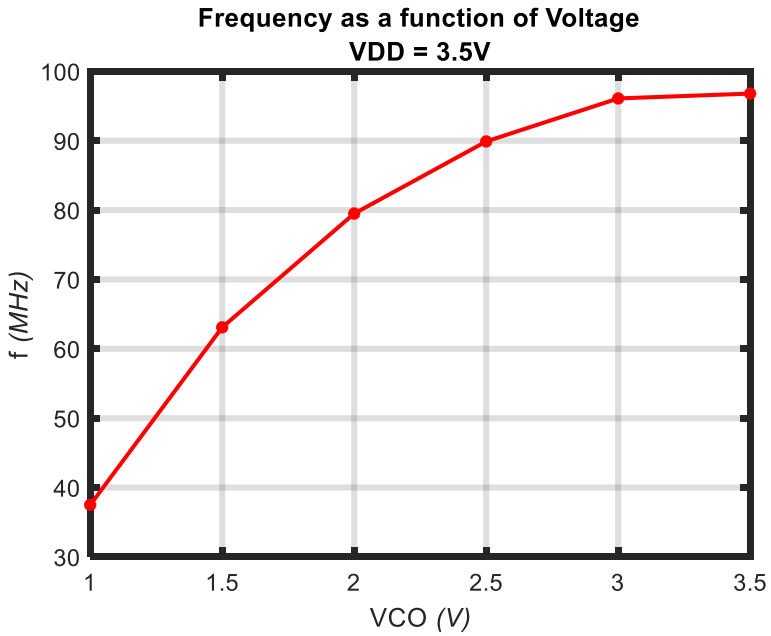
Minimum Voltages: VCO = 0.7V
*Rise time measured from
10 to 90% of VDD

VDD (V)	f_s (MHz)	Rise Time (ps)
2V	12.4	119
3V	13.8	96
4V	14.4	113
5V	15.0	101

Maximum Voltages: VCO = VDD
*Rise time measured from
10 to 90% of VDD

VDD (V)	f_s (MHz)	Rise Time (ps)
2V	49.9	86
3V	84.3	61
4V	106.5	56.2
5V	120.9	56

Looking at 3.5V, Plotting $f(VCO)$:



Linear Increase of VCO vs. Frequency

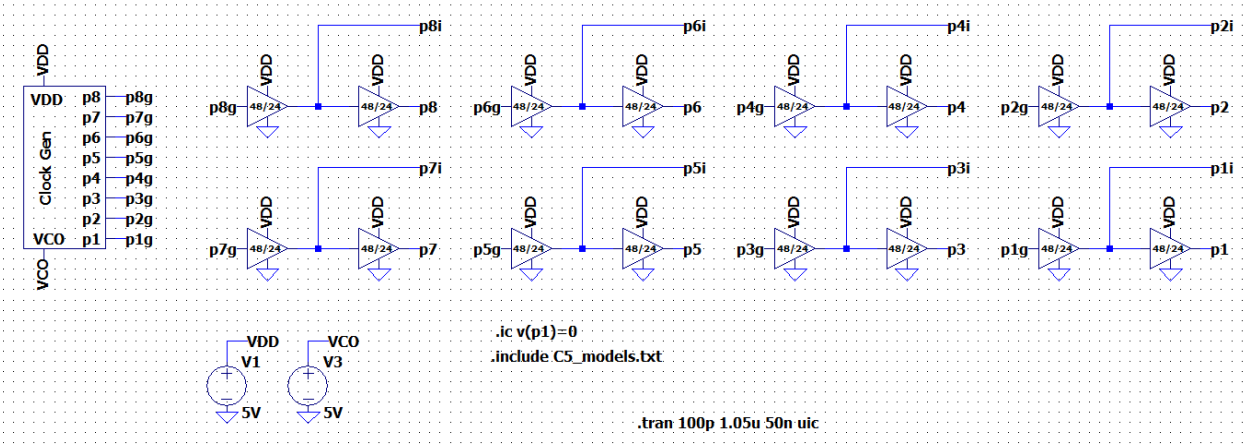
For this scope of the class, we will choose a conservative clock speed which will help with the design. The clock frequency, f_s, will be 100MHz at the following parameters:

VDD	3.7V	4V	4.5V	5V
VCO	3.6V	2.9V	2.85	2.85V

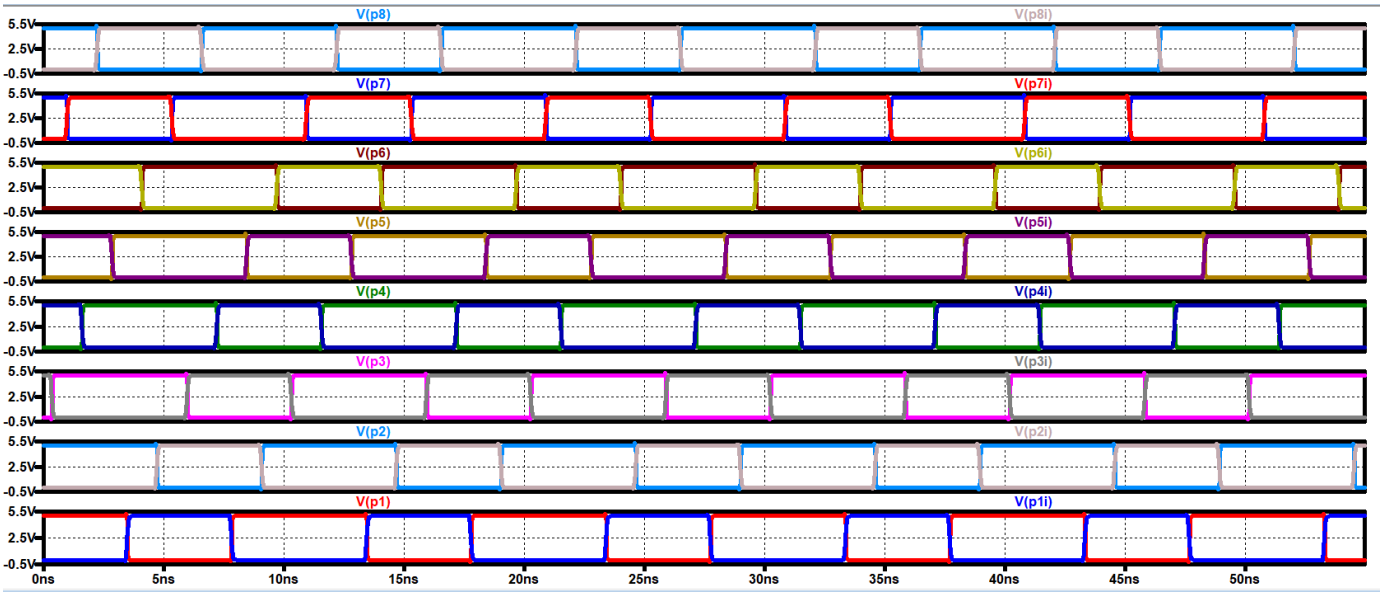
With changes in VDD, we can say that with a VCO voltage of 2.85V, we will get around 100MHz.

**Added Circuitry:

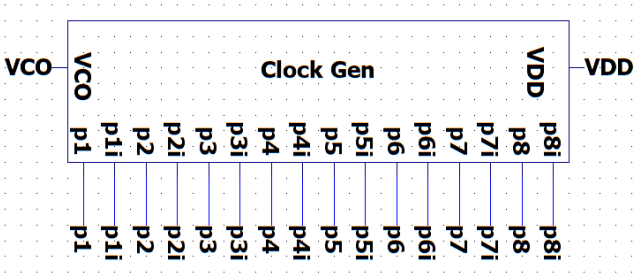
The following was added to the clocked comparator:



Output



Symbol:



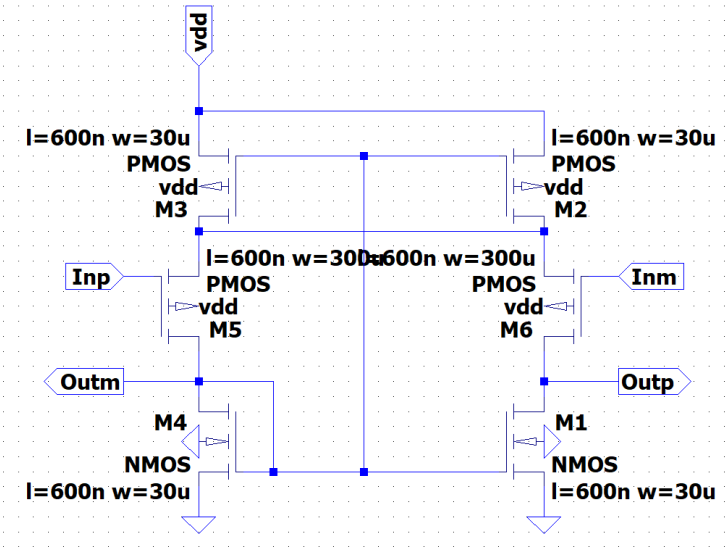
Note the analysis of the different VDD and VCO voltages are the same, the only added benefits are that the output of the clocks are all strengthened by a 48/24 inverter.

Differential Amplifier:

The next part that will be needed is the differential amplifier. This will be used in the integrator in the feedforward path of the ADC.

For this, we will design a simple differential amplifier that will be able to have a share between high gain (good for looking at small signals) and high speed (good to make it easier on the clocked comparator).

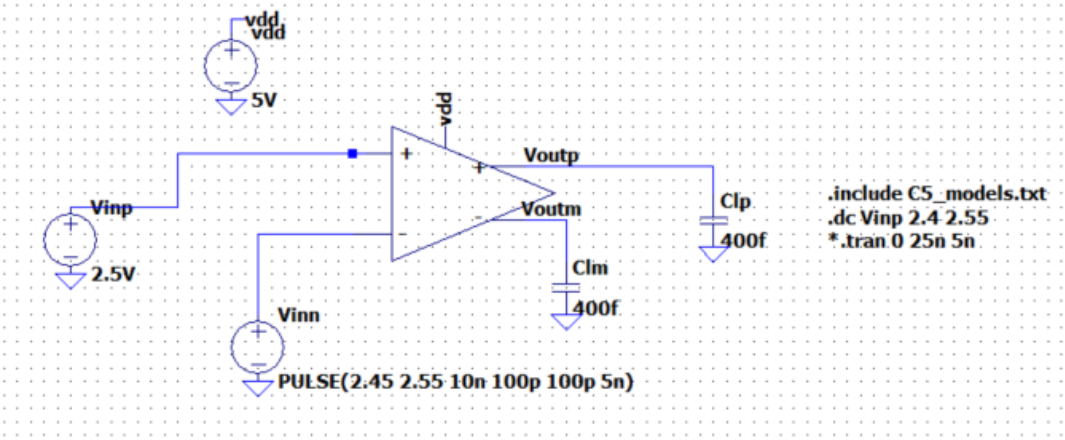
The first design will be a self-biased differential amplifier.



To test for the gain and the speed of this amplifier, we will have the following 2 test setups.

- DC Sweep of the input, looking at the change at the output.
- AC Analysis, looking at the open loop gain and phase margins.

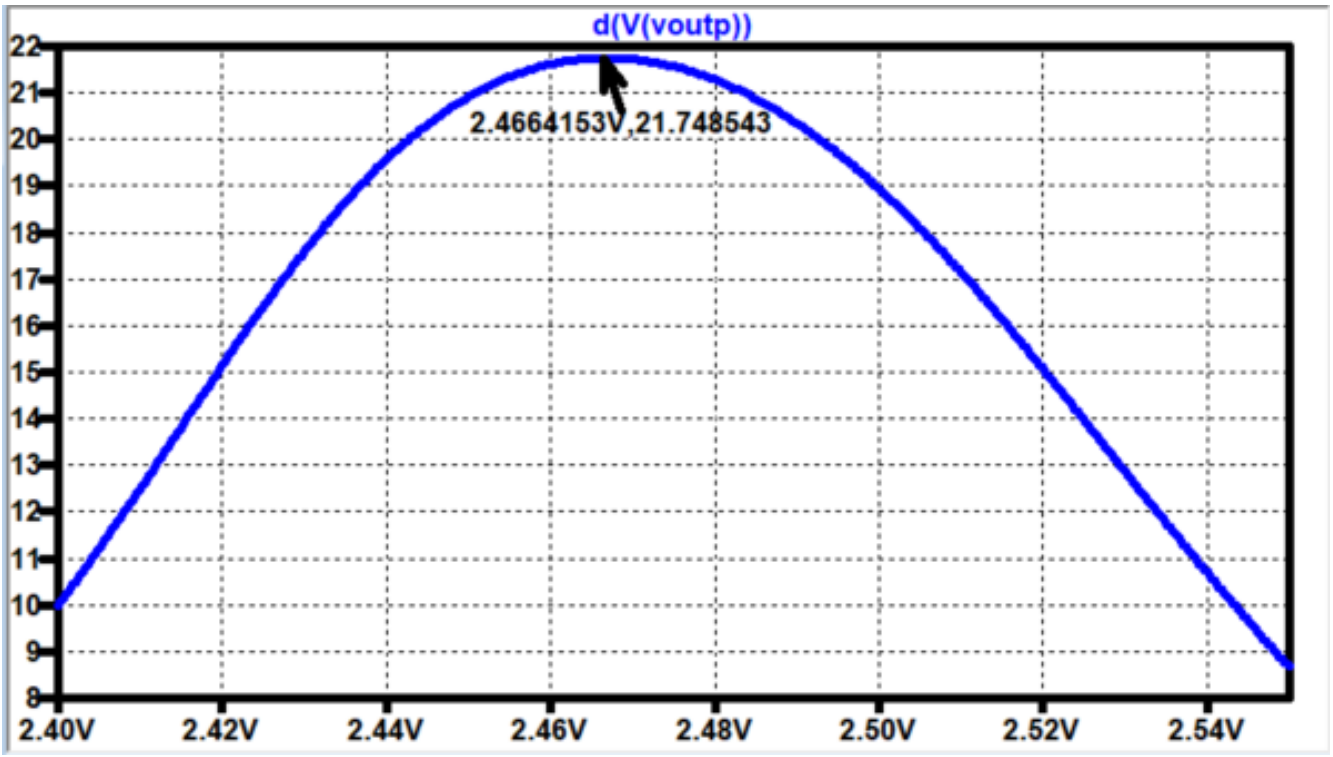
Test 1: DC Sweep



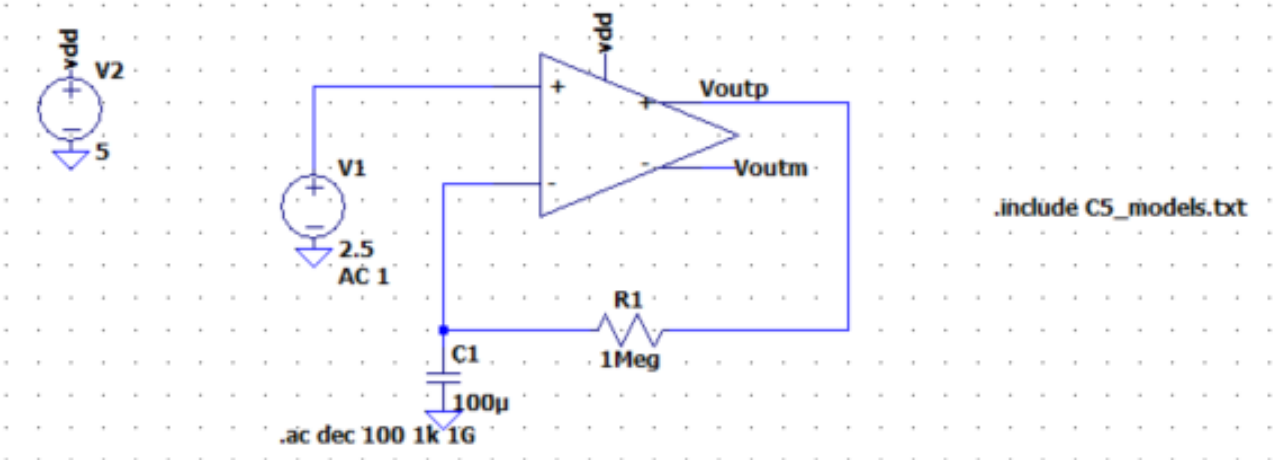
For the DC sweep, the positive terminal of the amplifier will be sweeped near the common mode voltage.

For the case of VDD = 5V, V_{CM} = 2.5V, and the gain is the derivative of the output with respect to the input.

Output: Gain = 21.7 (linear)

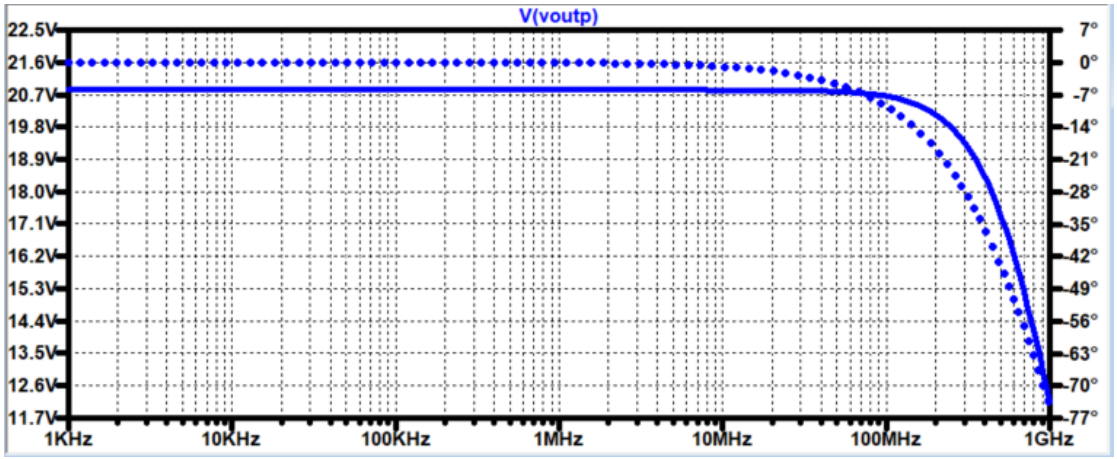


For both the gain and speed, we can perform an AC analysis on the amplifier:



The above circuit will be able to find the AC response and see the open loop gain with respect to frequency.

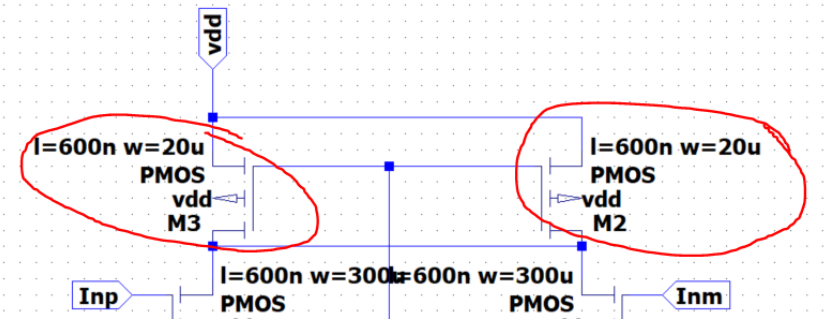
Output:



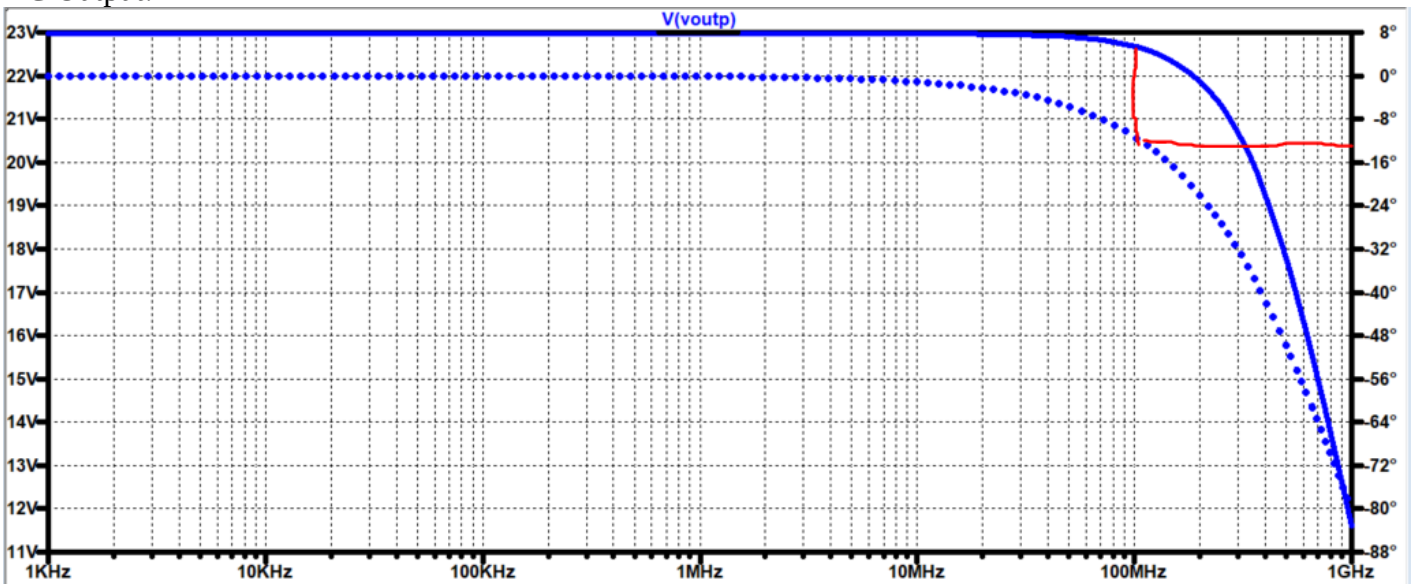
The open loop gain (G = 21) for the amplifier is achieved up to a frequency of around 100MHz.

For our design, we would like to have higher gain so that the comparator doesn't need to be modified much.

The current in each of the branches will be decreased by sizing down the following:



AC Output:



We have a new gain of 23! However, our gain has decreased slightly at 100MHz and also extra phase is added.

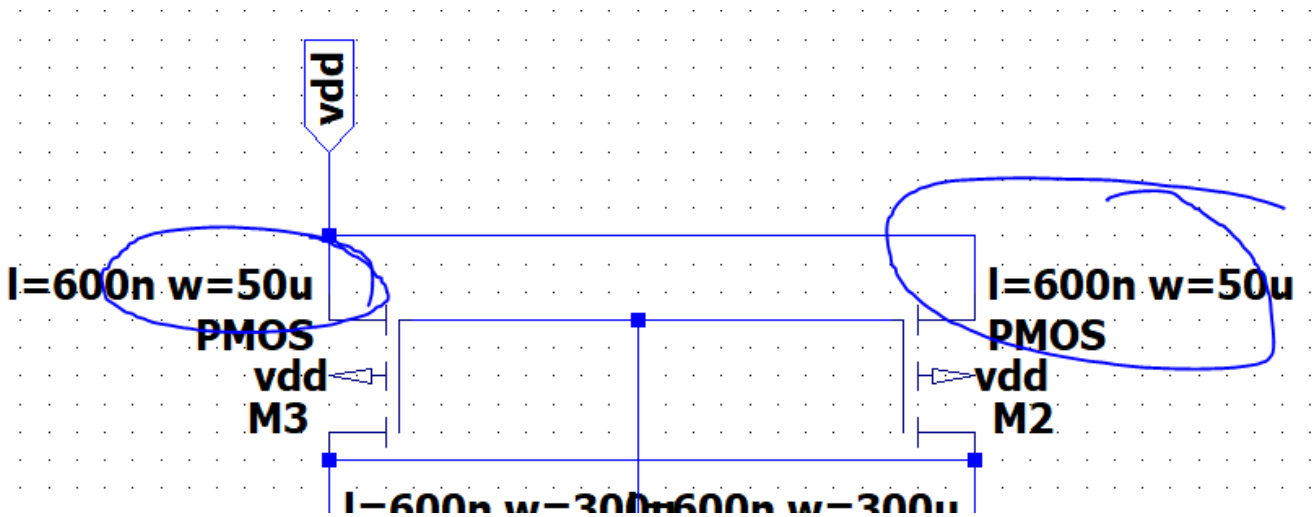
New Parameters Set:

Recall that the input of the integrator will see a max frequency change of:

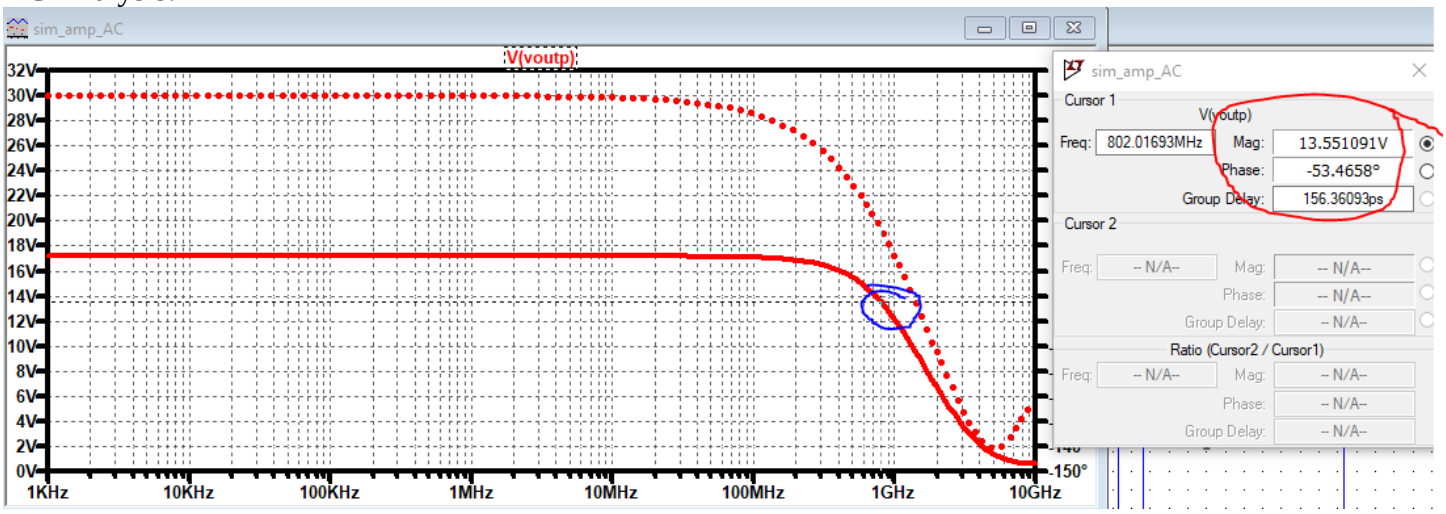
$$frequency\ at\ integrator\ input = K \cdot f_s$$

The feedback paths are all dumping new data into the integrator at K times the sampling frequency. Our integrator should be very fast in order to process all of this data.

For that, the PMOS current mirrors have the following change:



AC Analysis:



Although we will have a lower gain near the higher sampling rate, the increased speed will help with the integrator keeping up with the circuit.

Also note that the phase margin of the differential amplifier is around $-53°$.

From the CMOS textbook regarding Data Converters (Ch. 30, pg 1046):

$$f_{3dB} = \beta \cdot f_{un}$$

Where f_{un} is the Unity Frequency of an op-amp.

From above,

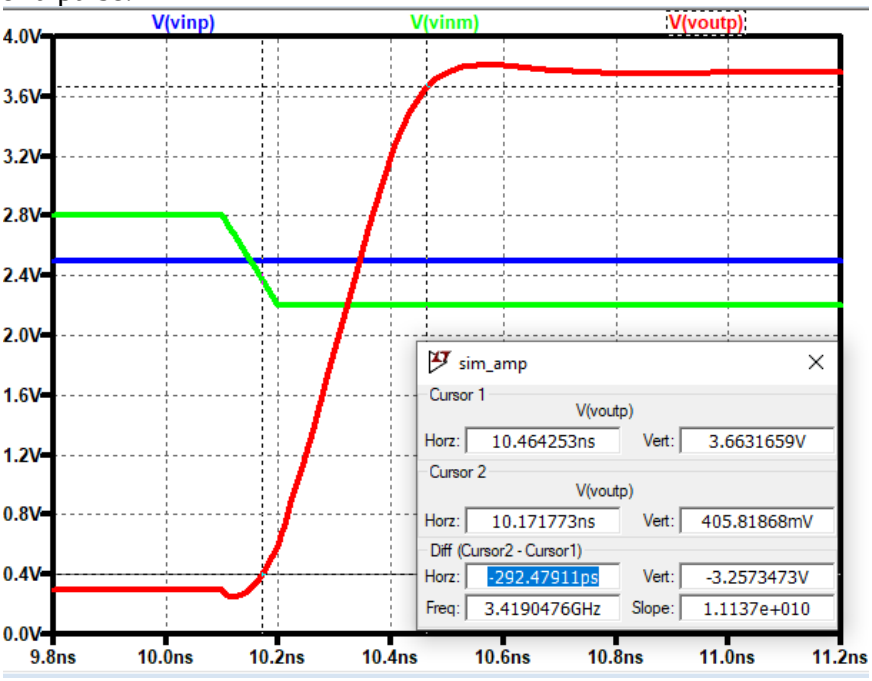
$$f_{un} = 5.82GHz, f_{3dB} = 604MHz$$

$$\beta = \frac{f_{3db}}{f_{un}} = \frac{.604}{5.82} = 0.103$$

The settling time can be found by:

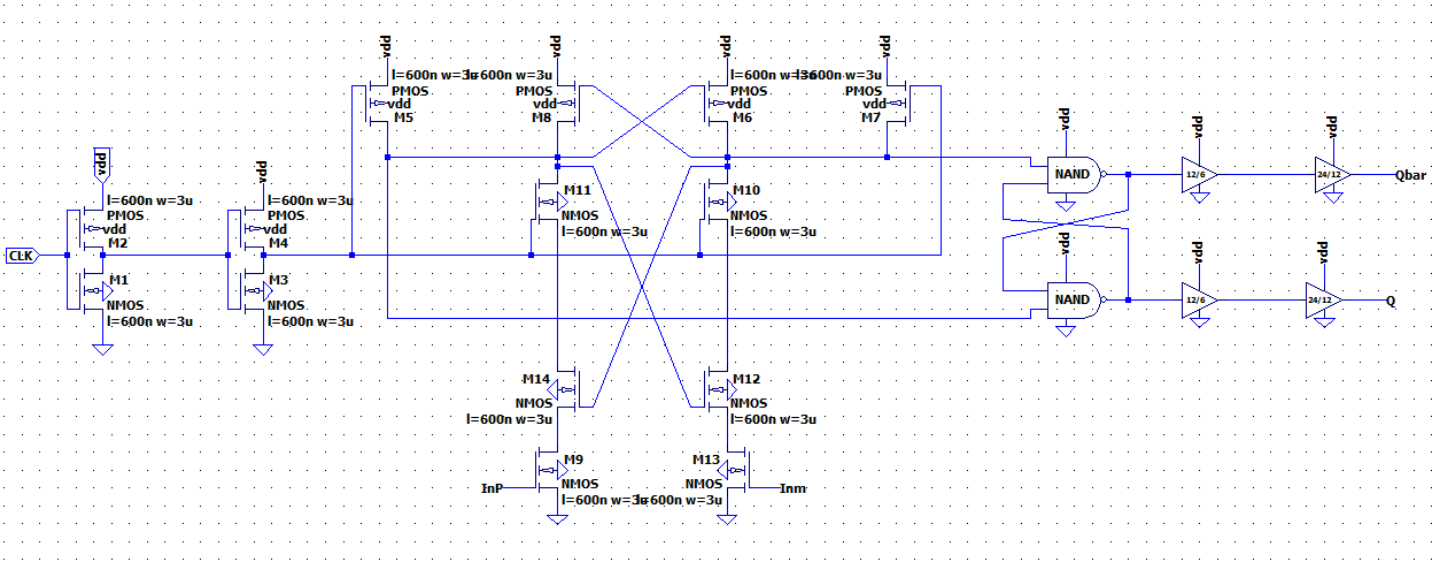
$$\tau = \frac{1}{2\pi f_{un}\beta} = \frac{1}{2\pi(5.82 \cdot 10^9)(0.103)} = 263ps$$

Looking at the output of a pulse:



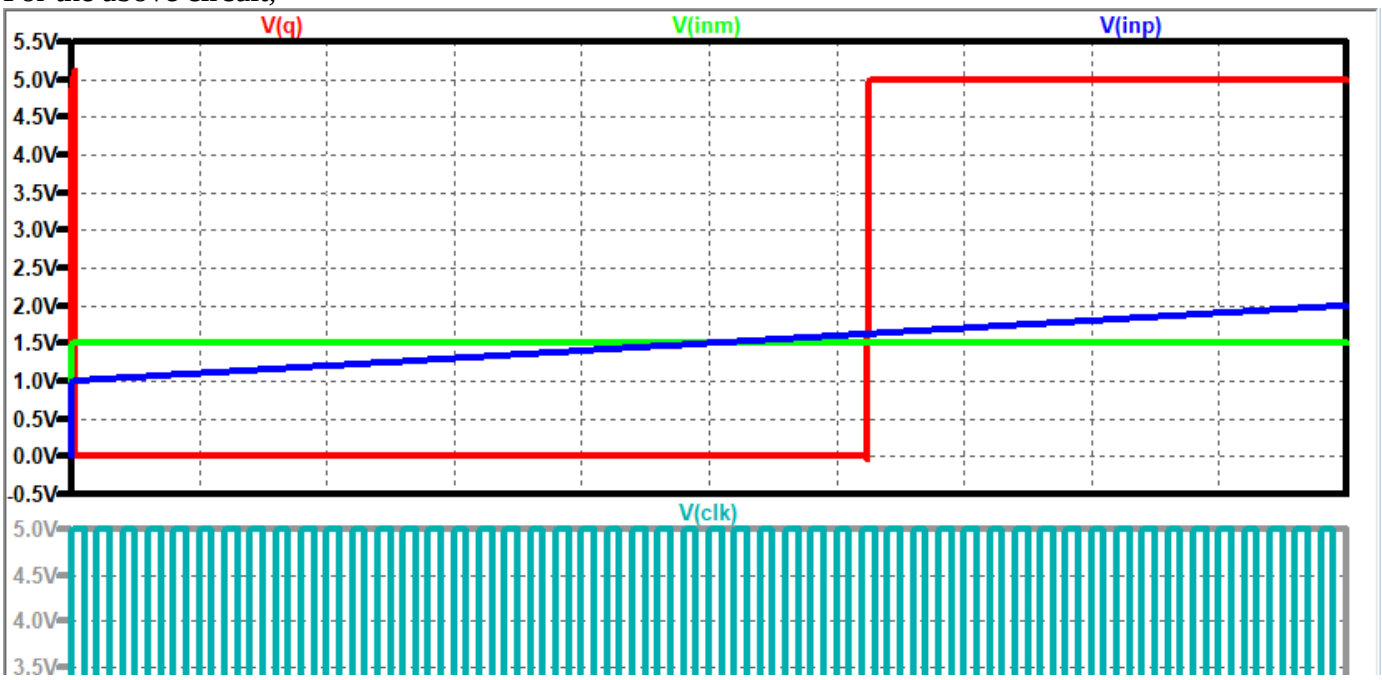
We can see that our edges match our hand calcs.

Clocked Comparator:



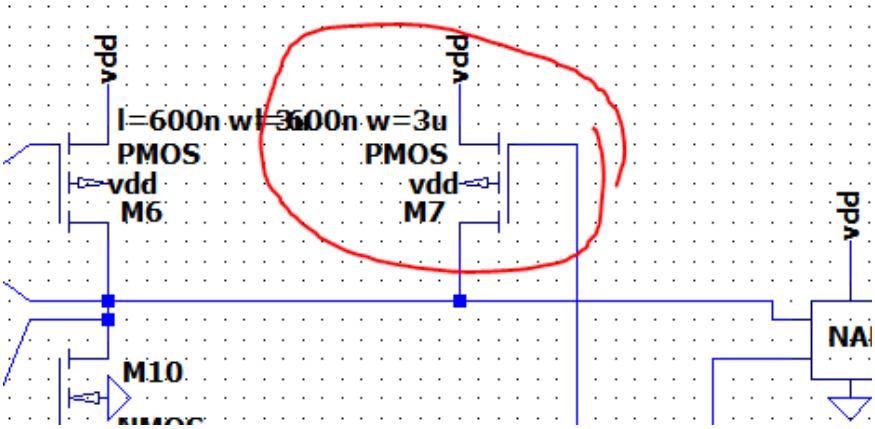
For the clocked comparator, the one parameter that can be changed is the switching point of the comparator.

For the above circuit,

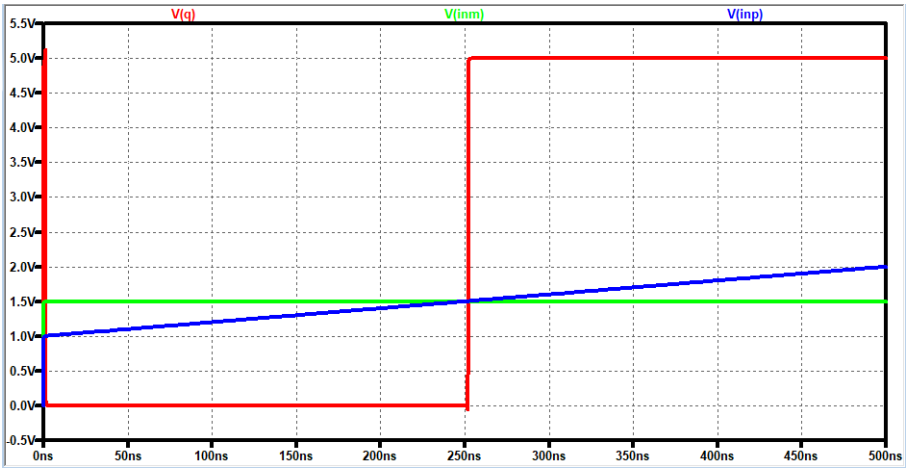


Our comparator has an offset!

We can fix this by changing the following device:

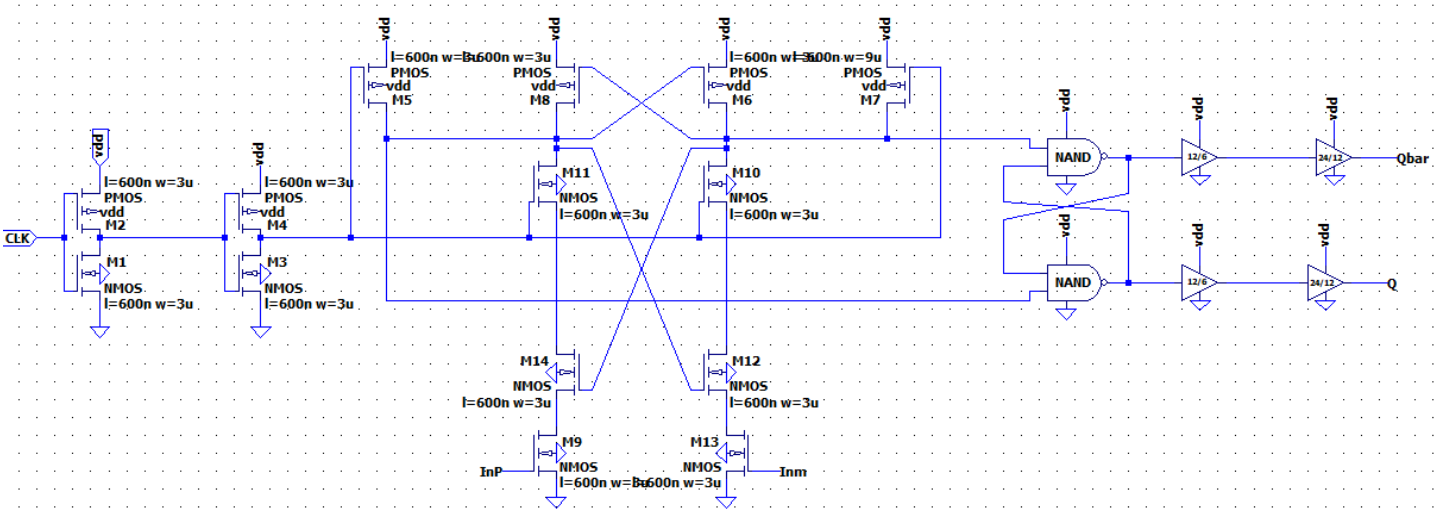


New output:



We now have a more balanced comparator.

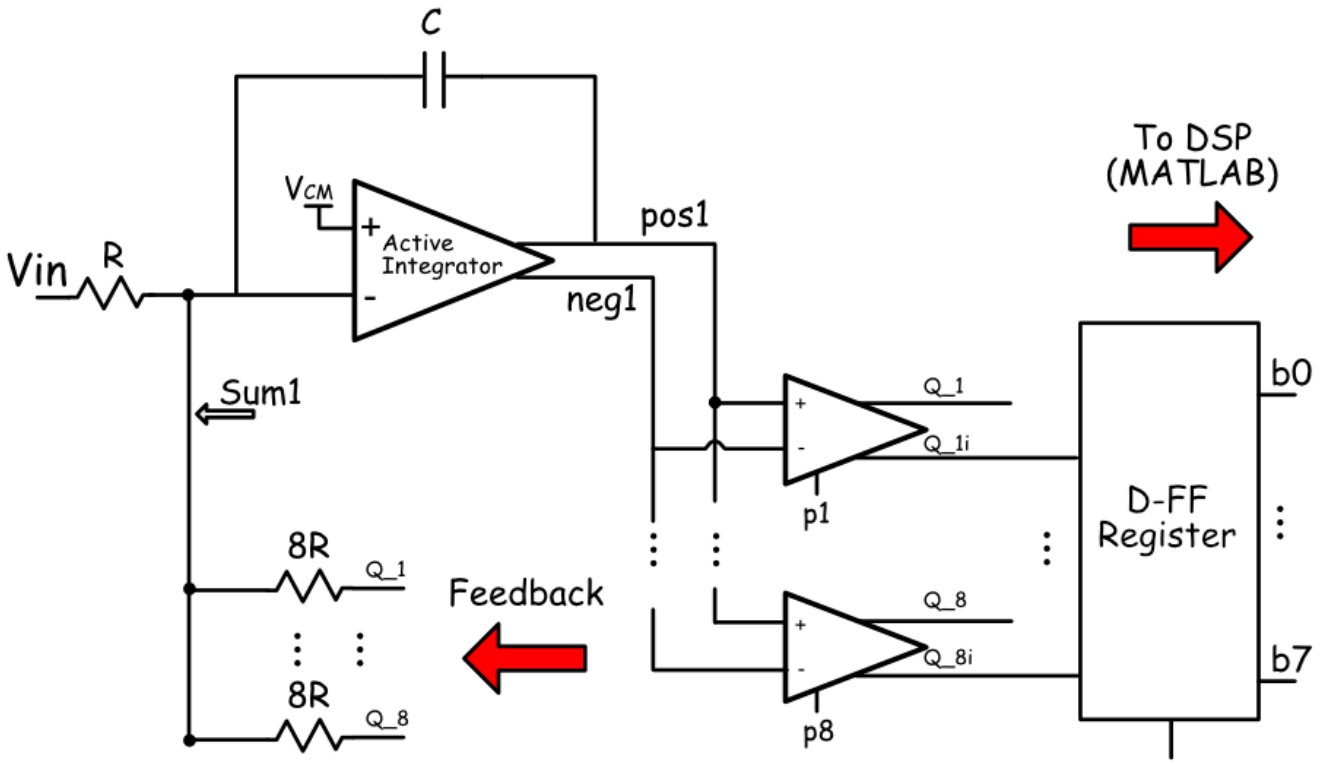
Final comparator circuit:



The Initial Design:

For the first design, we will implement a first-order active noise-shaping modulator with pure feedback resistors.

1st Order Noise Shaping Modulator w/Large Resistors:



Tests Performed (1st Order NS Modulator w/Large Resistors)

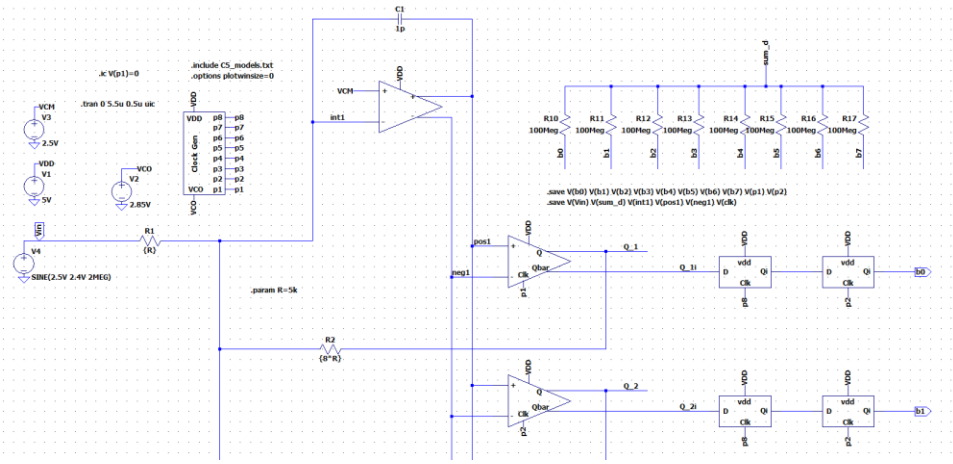
VDD	Amp	f_{in}	R	C	VCO/Freq.	OSR	SNR (Parallel)	SNR (Series)	NEFF (Parallel)	NEFF (Series)
5V	2.4V	2MHz	5kΩ	1pF	2.85V/100MHz	64	33.66	33.78	5.3	5.32
5V	2V	2MHz	6kΩ	1pF	2V/80MHz	64	37.13	37.65	5.87	5.96
5V	1.8V	2MHz	6kΩ	1pF	2V/80MHz	64	39.26	39.65	6.23	6.29
5V	1.5V	2MHz	6kΩ	1pF	2V/80MHz	64	34.46	35.05	5.43	5.53
5V	1.5V	1MHz	6kΩ	1pF	2V/80MHz	64	33.19	33.85	5.22	5.33
5V	1.8V	1MHz	6kΩ	1pF	2V/80MHz	64	35.76	38.54	5.64	6.11

The first approach is to make sure that we can create a stable feedback loop, at the cost of SNR and effective bits.

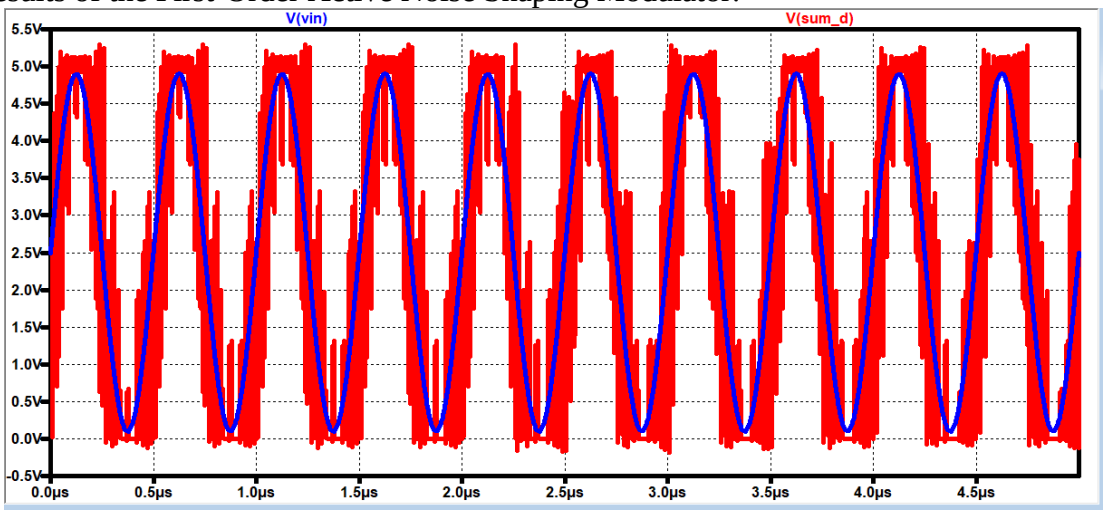
The topology above uses feedback resistors, all sized up to 8R. This is due to the K-Paths effectively sharing the output nodes and the input of the integrator (noted as Sum1). These resistors are in parallel and equate to a total feedback resistance of R so that the feedback resistance matches the input resistance at Vin.

The negative of this topology is that with faster input signal frequencies, the clocked comparator output will have a higher chance of holding irrelevant information. For example, the output of the first clocked comparator may not be relevant to the output of the later clocked comparators.

Small portion of the First-Order Active Noise Shaping Modulator (R = 3kΩ, C=1pF):

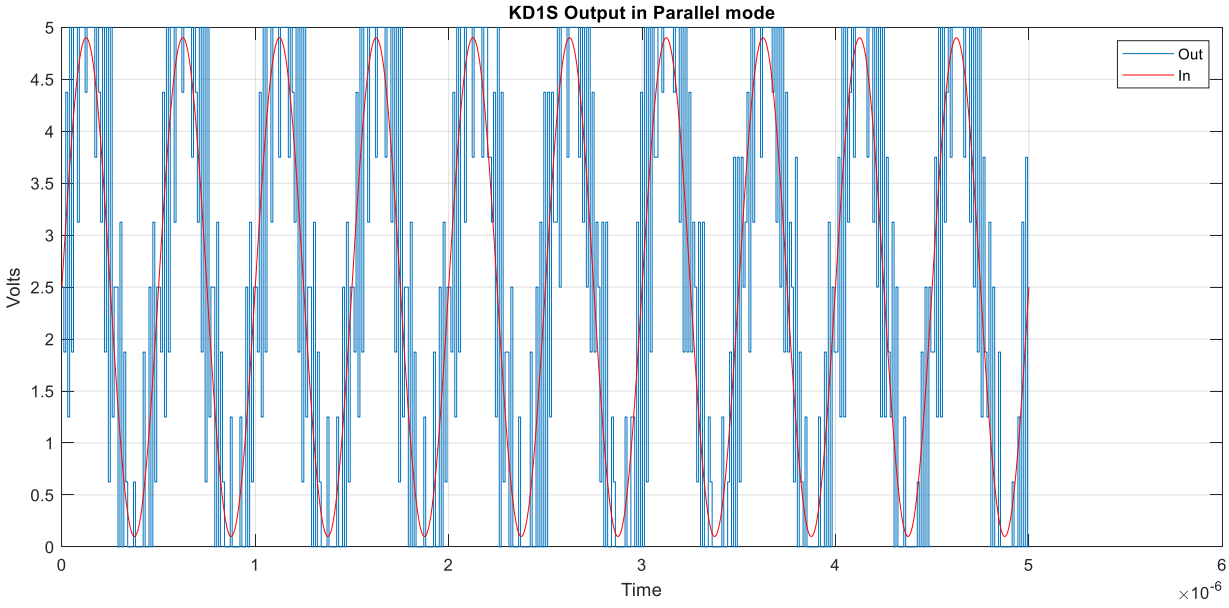


Simulation results of the First-Order Active Noise Shaping Modulator:

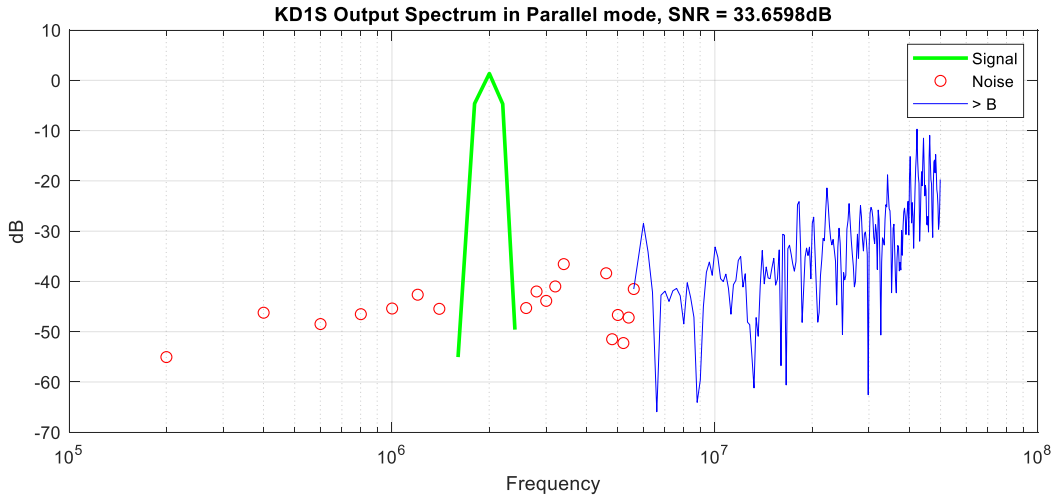


Data inputted into MATLAB (data successfully transferred):

Filtered output via MATLAB DSP (Parallel):



SNR (Parallel):



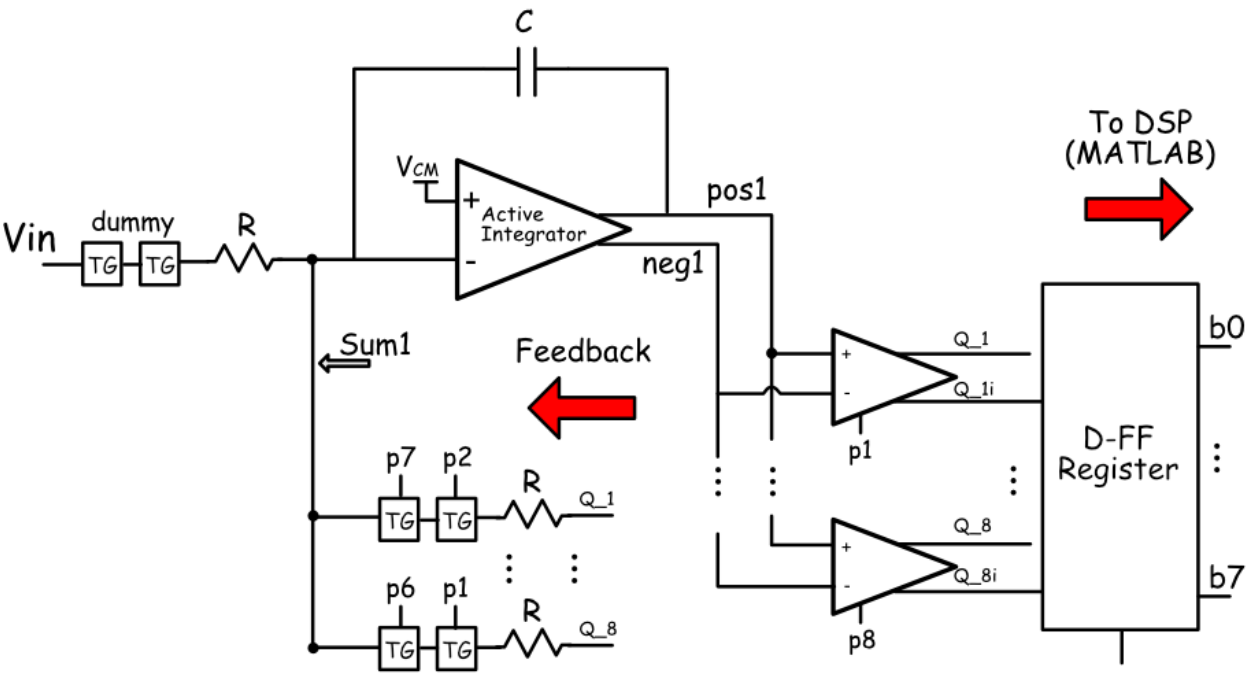
Effective Number of Bits (Parallel) = 5.30 bits
SNR value (Parallel) = 33.66dB

```
Kpath = 8 OSR = 64
For 1-bit output (serial mode) at Kpath*fs = 801 MHz
SNR = 33.78, Neff = 5.32, B = 6.26 MHz
For Kpath-bits output (parallel mode) at fs = 100 MHz
SNR = 33.66, Neff = 5.30, B = 6.25 MHz>>
```

Recall that for this topology, we have continuous dumping of data from all of the outputs of the clocked comparators.

We can change the above topology and use transmission gates, so that we are able to use the data of a clocked comparator for a short duration of time.

1st Order Noise Shaping Modulator w/Transmission Gates:



Tests of the 1st Order NS Modulator w/Transmission Gates:

VDD	f_{in}	Amp.	R	C	VCO/Freq.	OSR	SNR (Parallel)	SNR (Series)	NEFF (Parallel)	NEFF (Series)
5V	2MHz	2.4V	5kΩ	1pF	2.85V/100MHz	64	29.68	29.21	4.63	4.56
5V	2MHz	1.8V	6kΩ	1pF	2V/80MHz	64	35.81	34.62	5.65	5.45
5V	2MHz	1.8V	6kΩ	2pF	2V/80MHz	64	31.16	34.62	4.88	5.45
5V	2MHz	1.8V	6kΩ	0.8pF	2V/80MHz	64	33.0	31.42	5.19	4.92

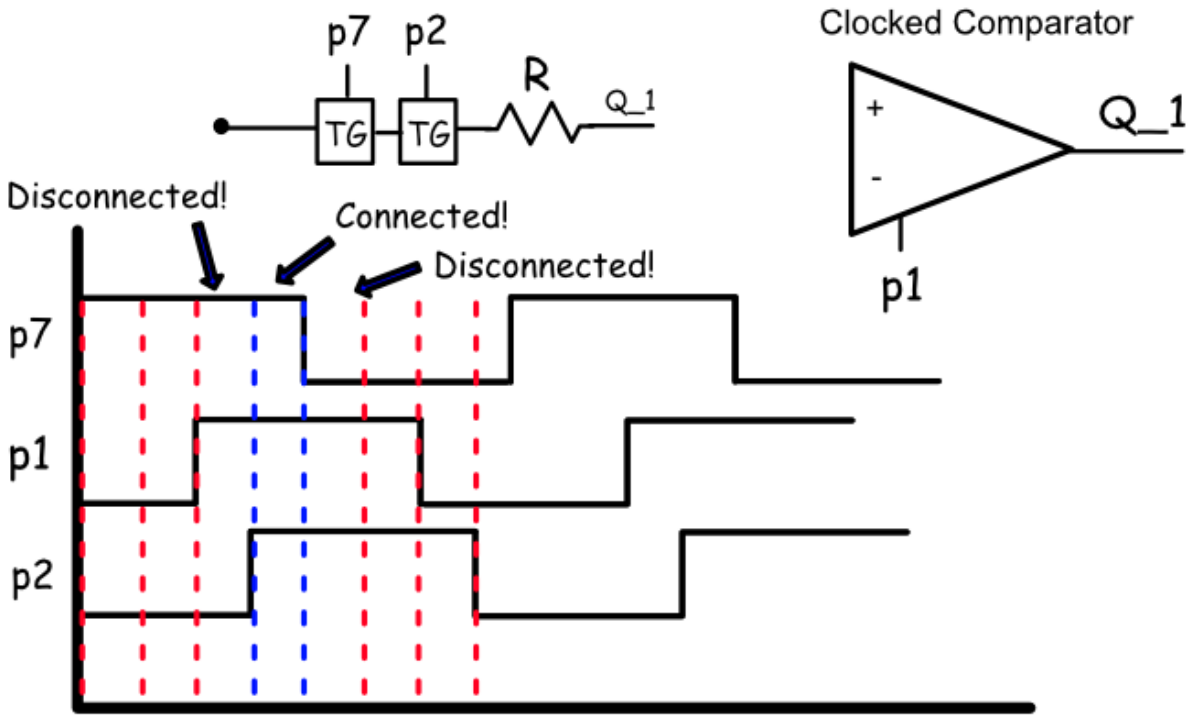
The transmission gates are implemented using an NMOS and PMOS with their drains and sources connected. 2 transmission gates are used. The idea is that the gates will be used to disconnect all but one of the feedback paths from the summation point of the integrator, so that the feedback paths can have the same resistance as the forward resistance from Vin.

2 things that need to be greatly considered for this topology:

- The amount of feedback that is being sent to the summation node
- The time that the transmission gates are open and closed.

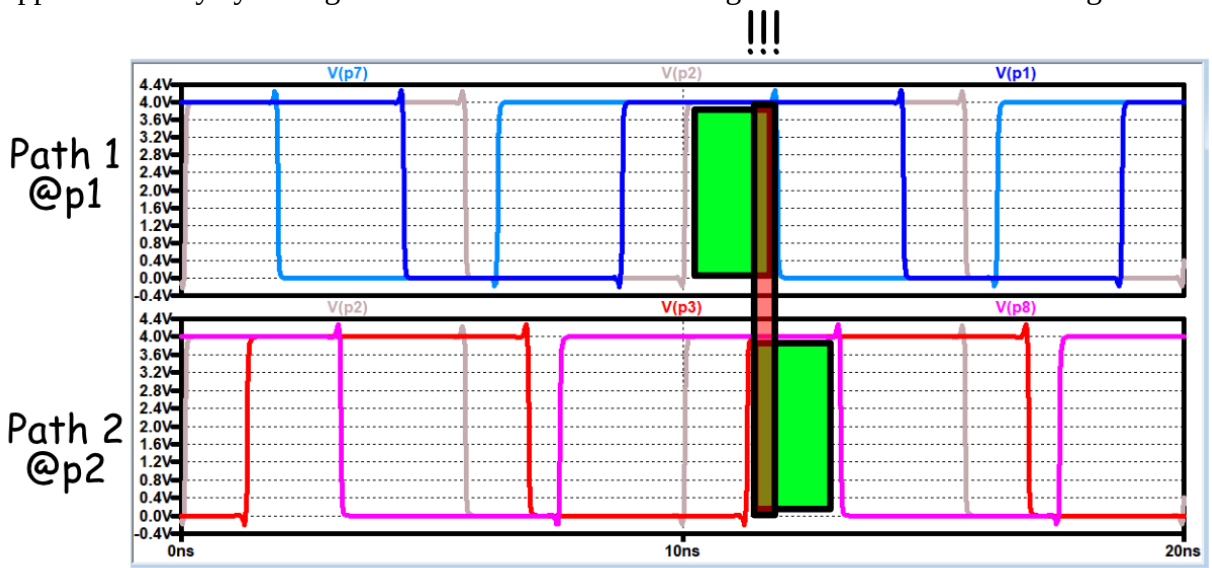
Ideally each branch should only be connected to the summation node for 1/8th of a clock phase.

The following figure will demonstrate the action in one path.



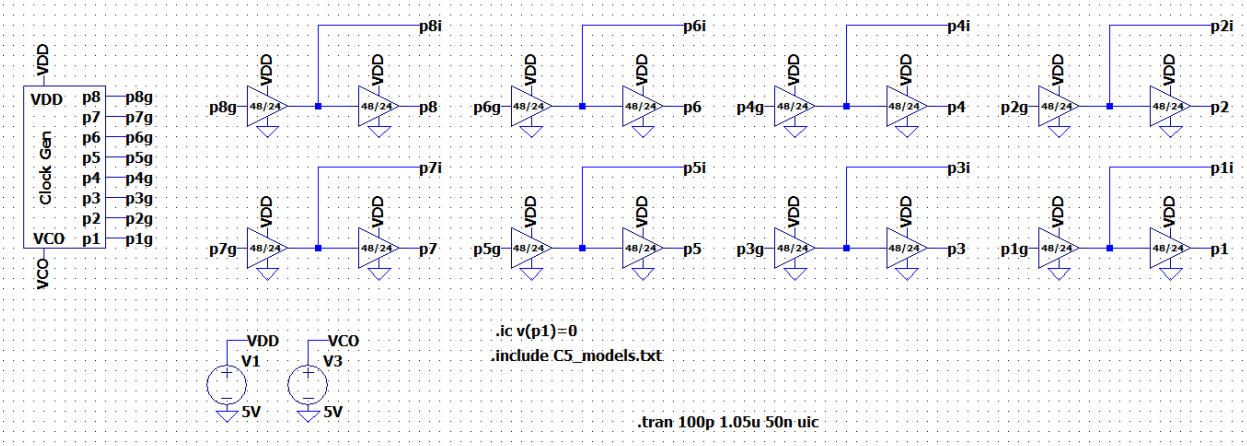
The transmission gates will ideally keep the feedback path connected for only 1/8th of a clock period (referenced to f_S).

Suppose the Duty cycle is greater than 50%. The following was taken from the Clock generator:



The area in green is where one of the paths are successfully connected to the summation node. With a high duty cycle (and using the clocks given), there is an area where at least 2 paths will be directly connected to the summation node!!!

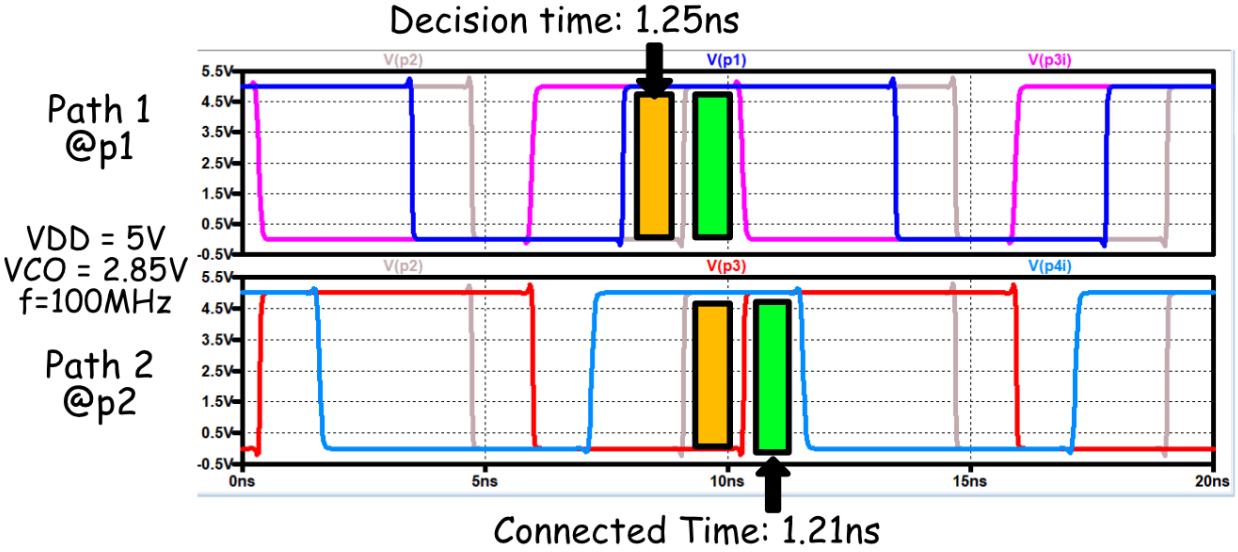
We can play a game by adding some buffed inverters and complimentary clocks.
Presenting the new clock generator:



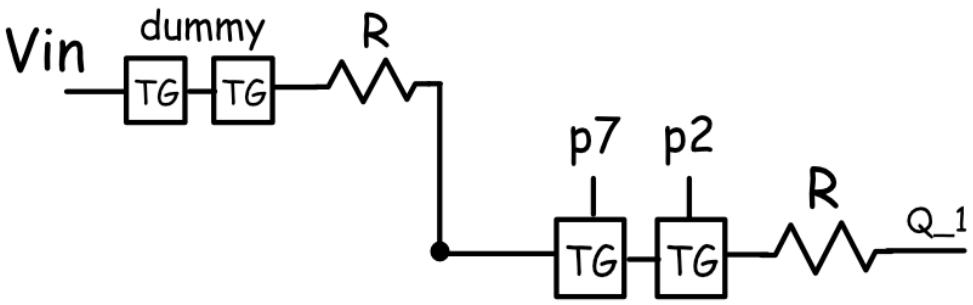
The output of the original clock generator had a 24/12 inverter. This new topology uses 48/24 inverters so that not only are we able to invert all of the clocks, but we also get strong output signals that will be able to carry the large capacitances of the Transmission gates!

We note that to get a lower duty cycle, either delay or inverting a high duty signal will give us lower duty. For this design, the high duty signals will be inverted to give us the lower duty cycle.

Playing a few games with the clocks, we see that if we want a low duty on clock P7, the opposite signal of P7 is P3, therefore inverting P3 will give us a mimicked low duty cycle to replace P7.



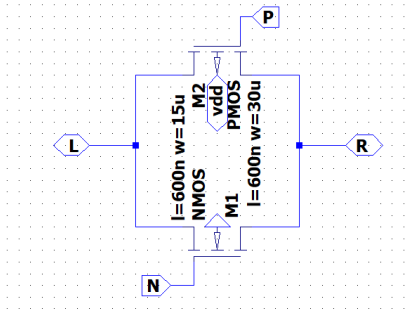
We see that our connection times are separated! Also due to the stronger signals, our clocks should still be able to carry the transmission gates.
The transmission gates will also contain an internal resistance. To compensate for this, a dummy transmission gate is placed at the input feedforward resistor to minimize resistor mismatch.



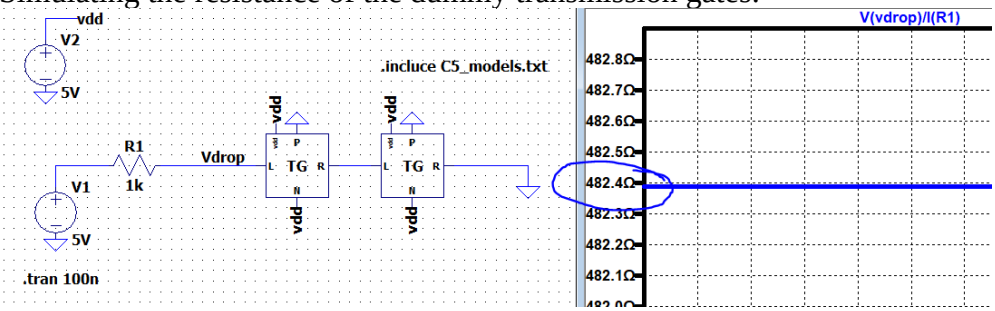
Transmission gates at Vin to match any resistance in the Feedback loop.

Also, the size of the Transmission Gates will need to be considered. Wide MOSFETs have less internal resistance, however, it introduces more capacitance at the gates, which means that the clock generator should be able to carry that load.
Buffing the clocks with a buffer inverter should be able to compensate for this increased load.

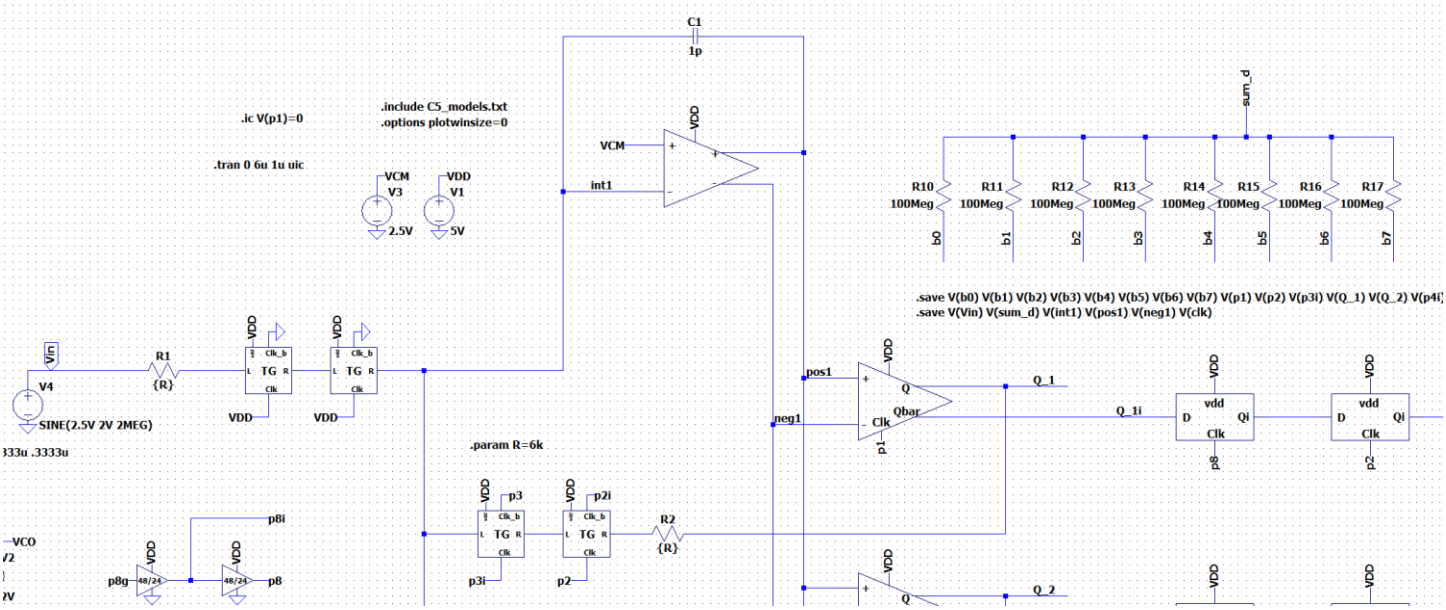
A Transmission Gate (W_N = 15µm, W_P = 30µm):



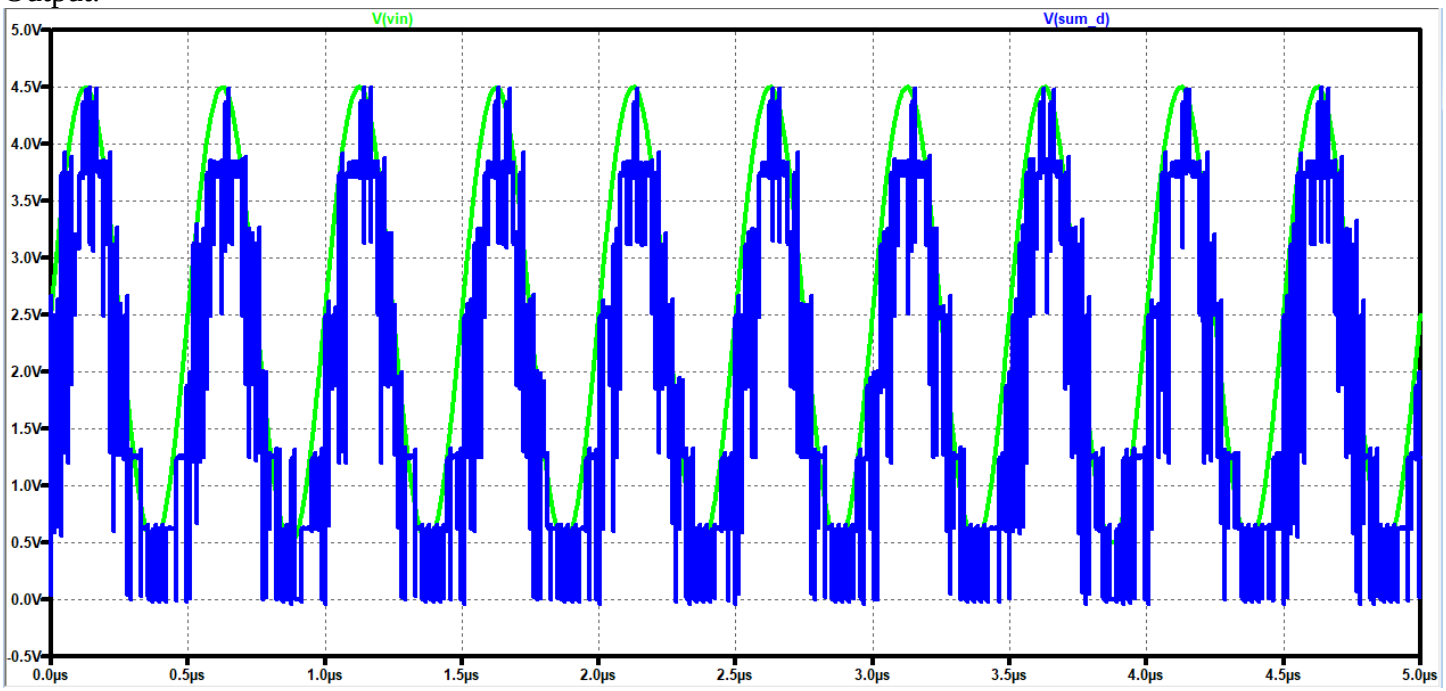
Simulating the resistance of the dummy transmission gates:



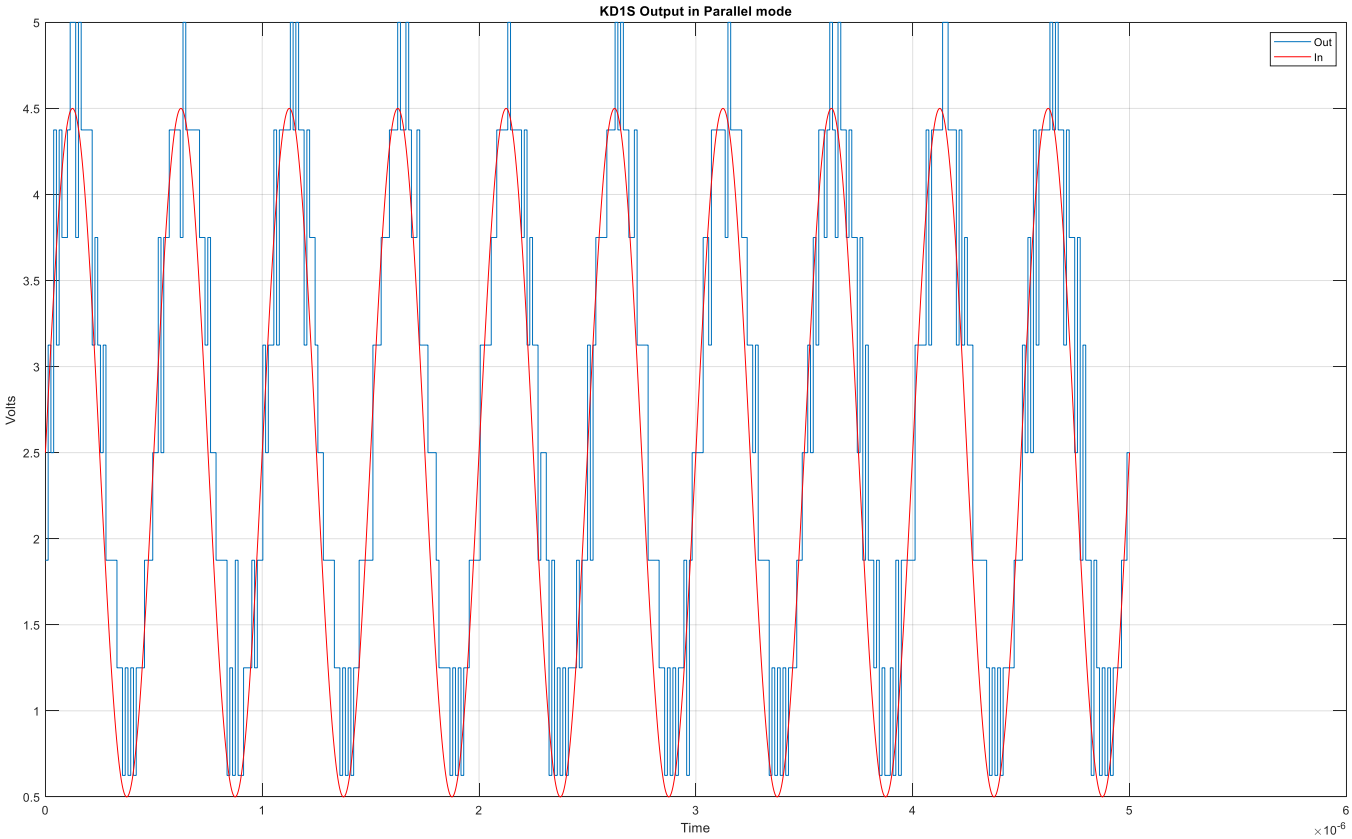
Our resistance of the 2 transmission gates is around 482Ω. This is a lot, however, if we increase the feedforward/feedback resistors, then this resistance can be relatively small.



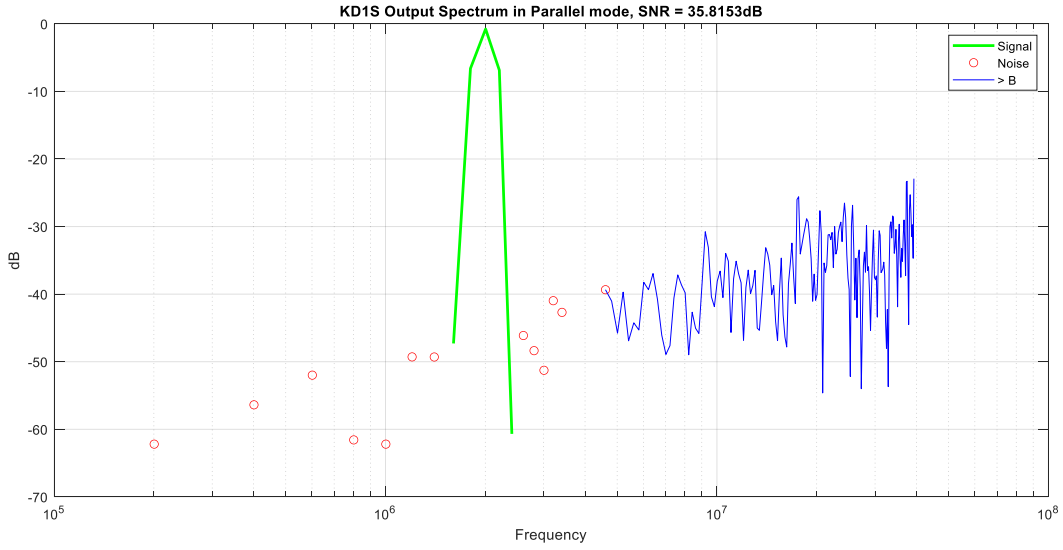
Output:



MATLAB Transfer of data:



SNR (Parallel): 33.84dB (Effective Bits = 5.65)



Hand-calculating SNR (1st Order):

Recall the SNR equation for a 1st order Noise Shaping Topology:

$$SNR_{ideal} = 6.02N + 1.76 - 20\log\frac{2\pi RC \cdot f_s}{\sqrt{12}} + 30\log K$$

With the following parameters:

$$R = 6k\Omega, C = 1pF, f_s = 80MHz, K = 8$$

Our Signal to Noise Ratio is:

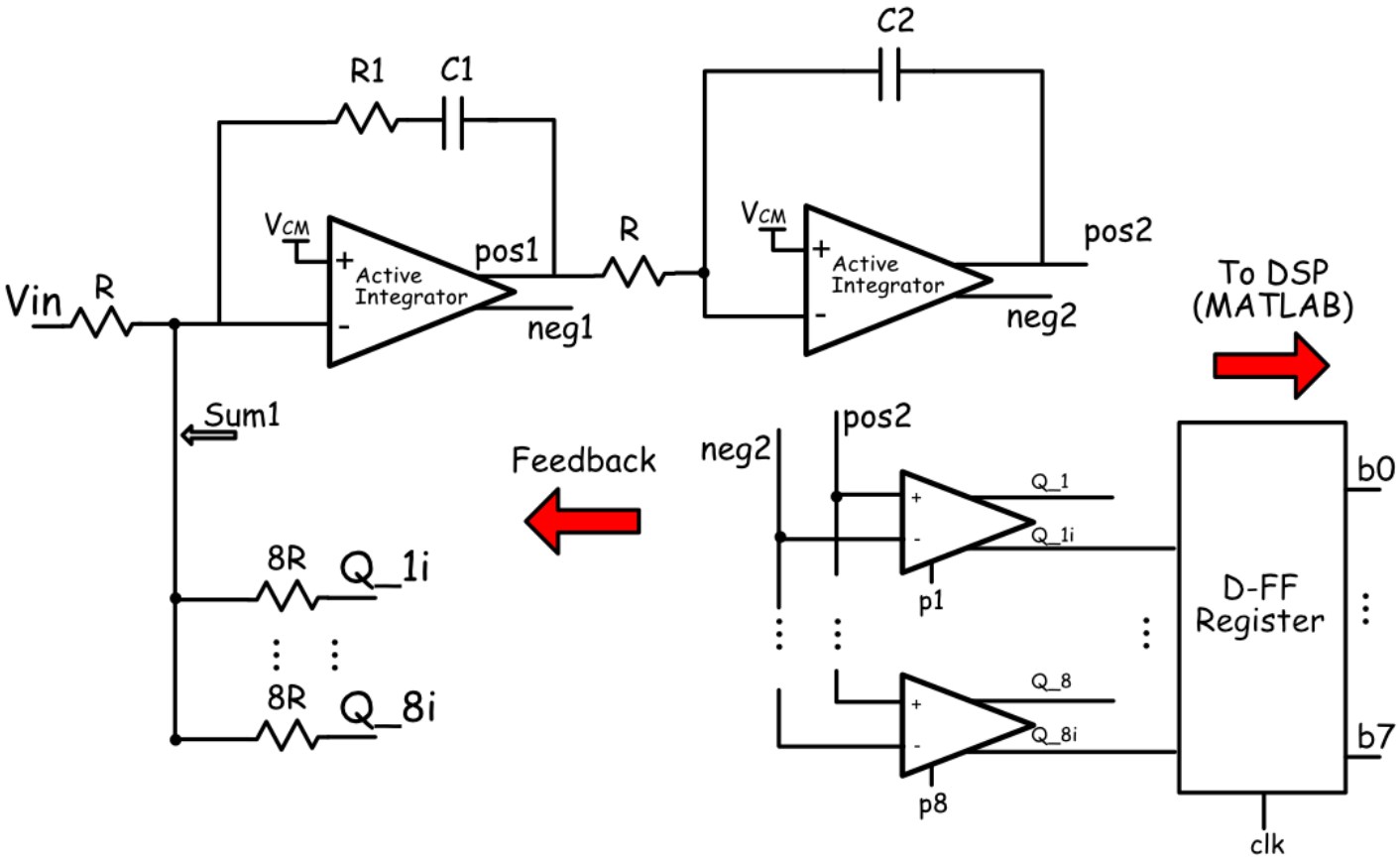
$$SNR_{ideal} = 6.02N + 1.76 - 20\log\frac{2\pi(6k \cdot 1p) \cdot 80M}{\sqrt{12}} + 30\log 8$$

$$SNR_{ideal} = 6.02N + 1.76 - 1.2 + 27.09$$

$$\boxed{SNR_{ideal} = 33.67dB}$$

2nd Order NS Modulator w/Large Resistors:

VDD	f_{in}	Amp.	R	R_1	C_1	C_2	VCO/Freq.	OSR	SNR (Parallel)	SNR (Series)	NEFF (Parallel)	NEFF (Series)
5V	2MHz	2.4V	5kΩ	5kΩ	6pF	1pF	2.85V/100MHz	64	30.53	30.49	4.78	4.77
5V	2MHz	2.4V	5kΩ	5kΩ	1pF	1pF	2.85V/100MHz	64	31.1	30.63	4.87	4.79
5V	2MHz	2.4V	5kΩ	5kΩ	1pF	2pF	2.85V/100MHz	64	28.83	29.12	4.49	4.54
5V	2MHz	1.8V	6kΩ	6kΩ	5pF	2pF	2V/80MHz	64	38.95	38.76	6.17	6.14



The theory for the above is that we will be using 2 integrators that will provide 2nd order noise shaping. In theory, the 2nd order noise shaping will provide less noise at the lower frequencies. The negative is that our bandwidth has to be limited to a lower frequency, else we will be taking in noise at higher frequencies.

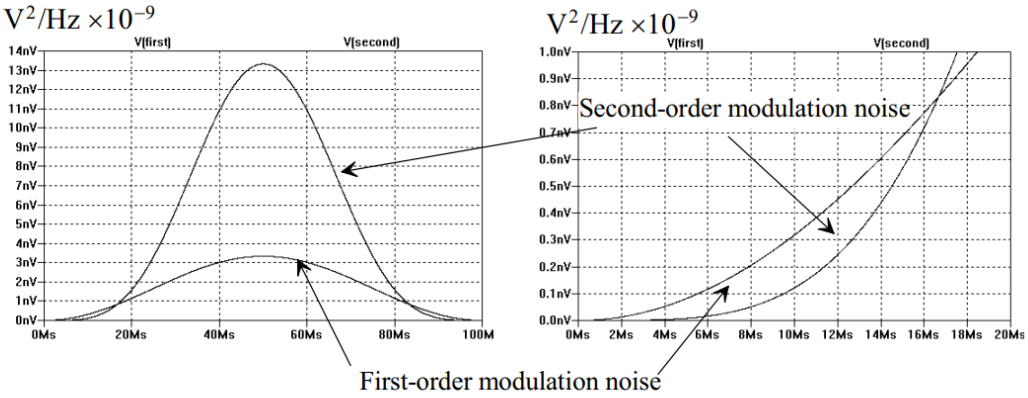


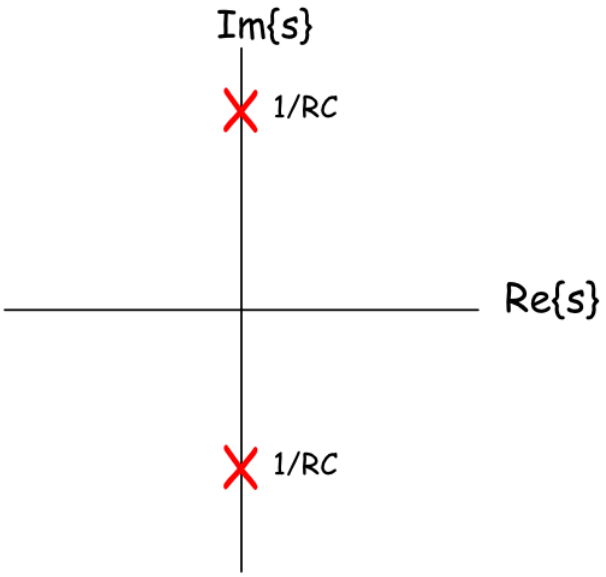
Figure 7.23 Comparing first- and second-order NS modulator's modulation noise.

The figure above shows that at lower frequencies, our noise is practically small, however, it increases by a squared term and can give unwanted noise at higher frequencies.

The second thing to note, there is no feedback path into the second amplifier. This is to make the design smaller (due to the large resistors), however, we will generate a pole!

Looking at the transfer function for the 2nd order modulator with no feedback path:

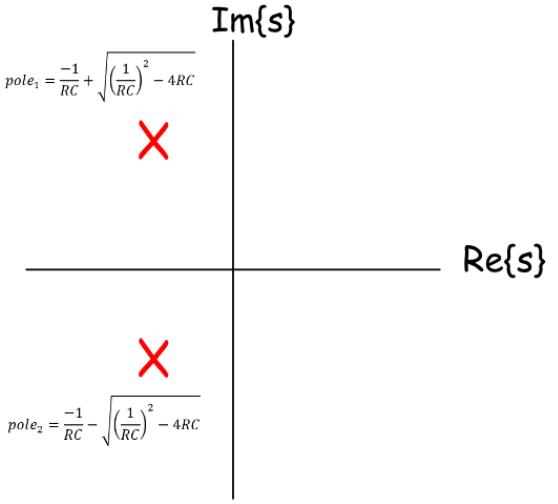
$$v_{out} = \overbrace{\frac{1}{(j\omega RC)^2+1}}^{STF} \cdot v_{in} + \overbrace{\frac{(j\omega RC)^2}{(j\omega RC)^2+1}}^{NTF} \cdot V_{Qe}$$



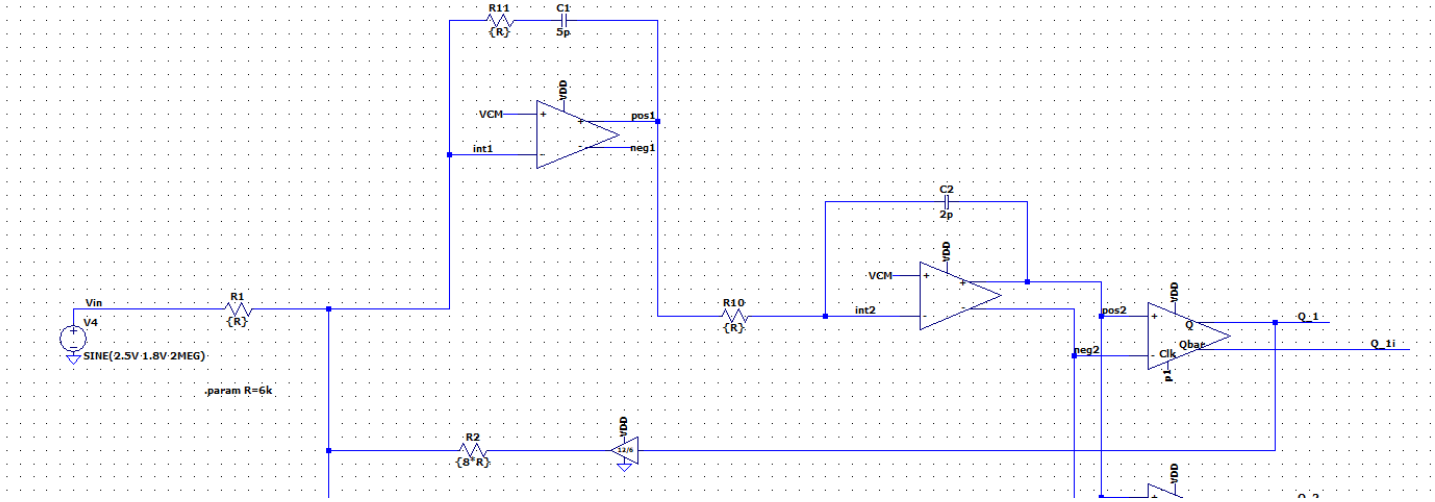
In the Laplace domain, the poles will be on the imaginary axis, which means that our system is on the verge of instability. To fix this problem, a resistor is placed in the 1st integrator path with the capacitor to move the poles away from the Right-hand plane.

New Laplace representation of the poles:

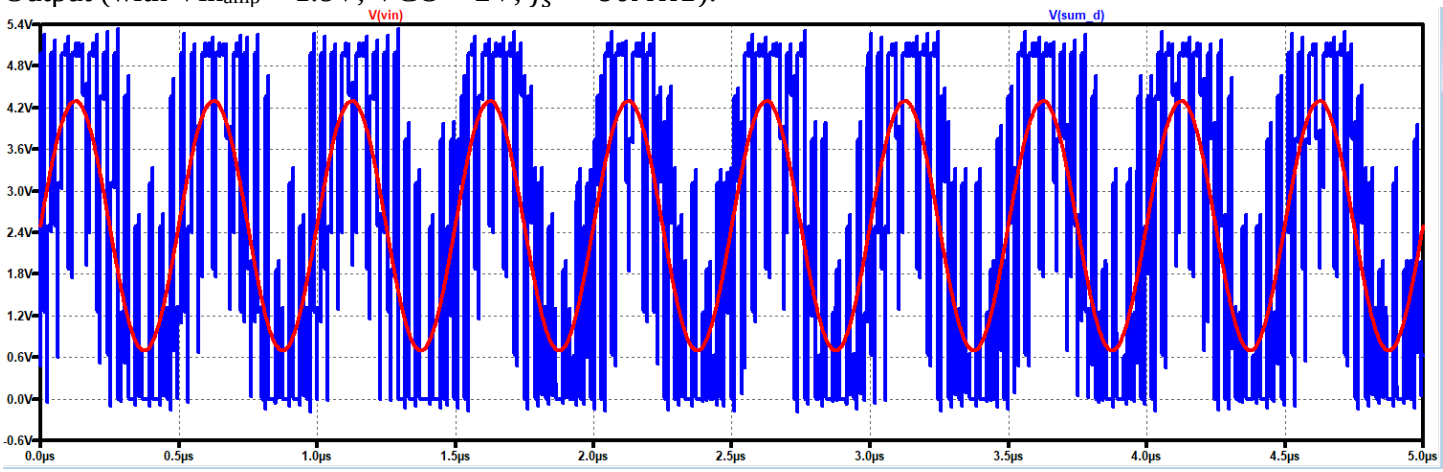
$$v_{out} = \overbrace{\frac{j\omega RC+1}{(j\omega RC)^2+j\omega RC+1}}^{STF}\cdot v_{in} + \overbrace{\frac{(j\omega RC)^2}{(j\omega RC)^2+j\omega RC+1}}^{NTF}\cdot V_{Qe}$$



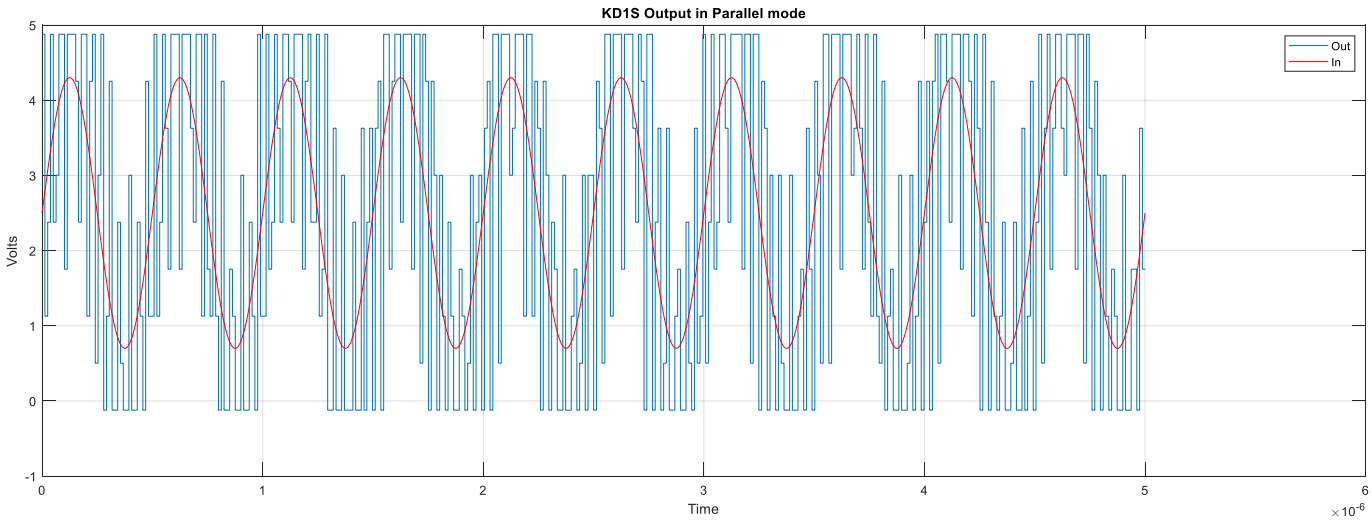
Small Portion of the 2nd order NS Modulator w/Large Resistors:



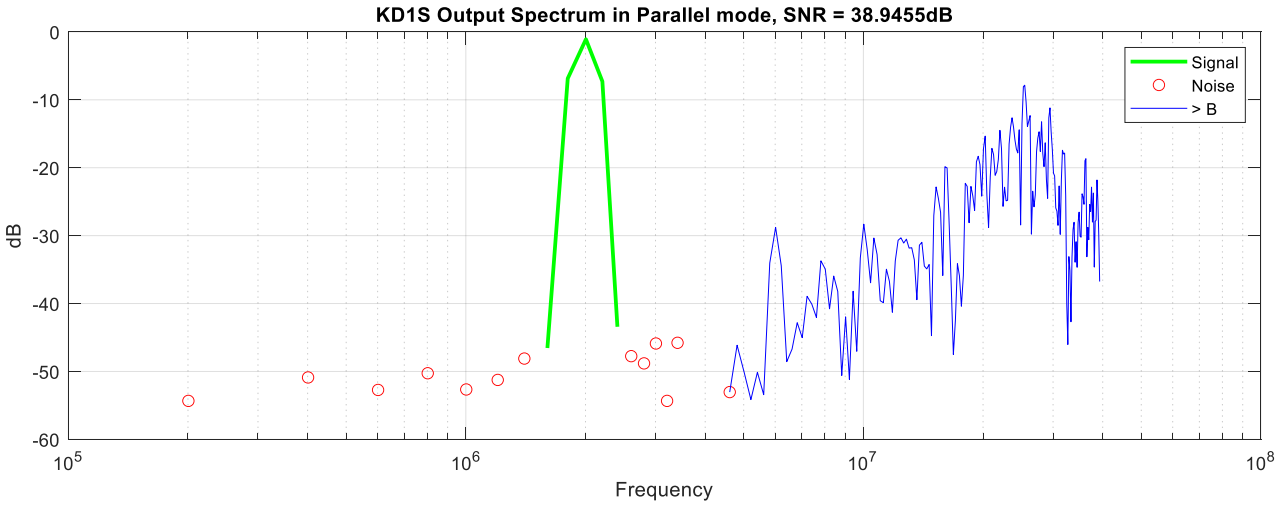
Output (with Vin_{amp} = 1.8V, VCO = 2V, $f_s = 80MHz$):



Successful transfer to MATLAB:



SNR (Parallel):

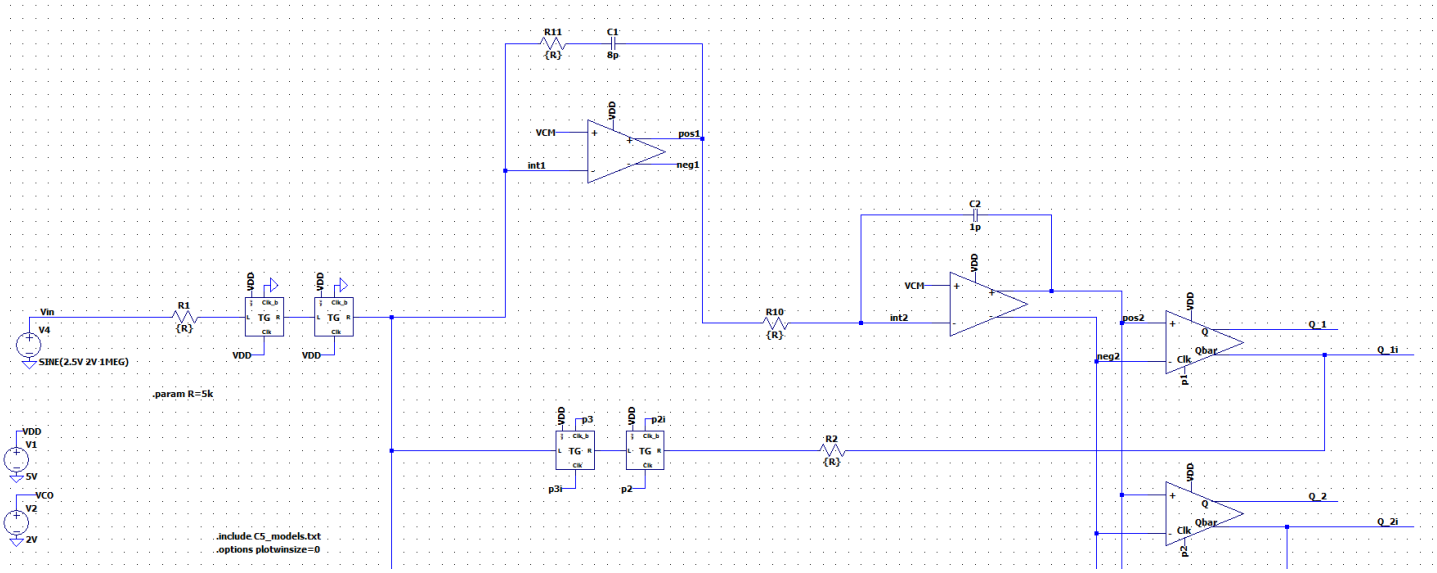


MATLAB Output:

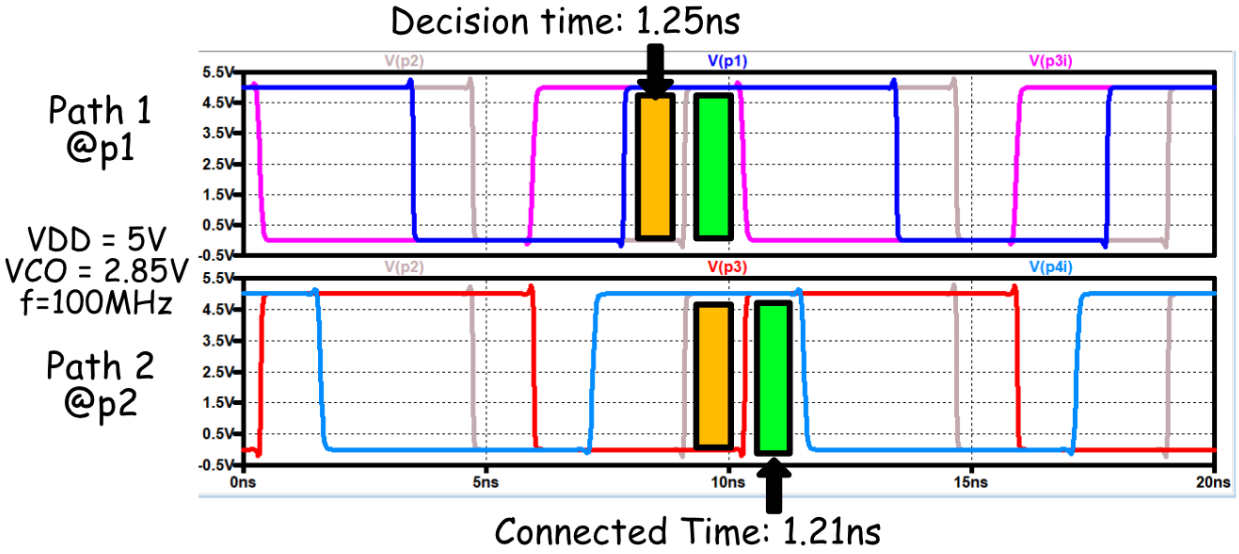
```
Kpath = 8 OSR = 64
For 1-bit output (serial mode) at Kpath*fs = 632 MHz
SNR = 38.76, Neff = 6.14, B = 4.94 MHz
For Kpath-bits output (parallel mode) at fs = 79 MHz
SNR = 38.95, Neff = 6.17, B = 4.92 MHz>>
```

2nd Order NS Modulator w/Transmission Gates:

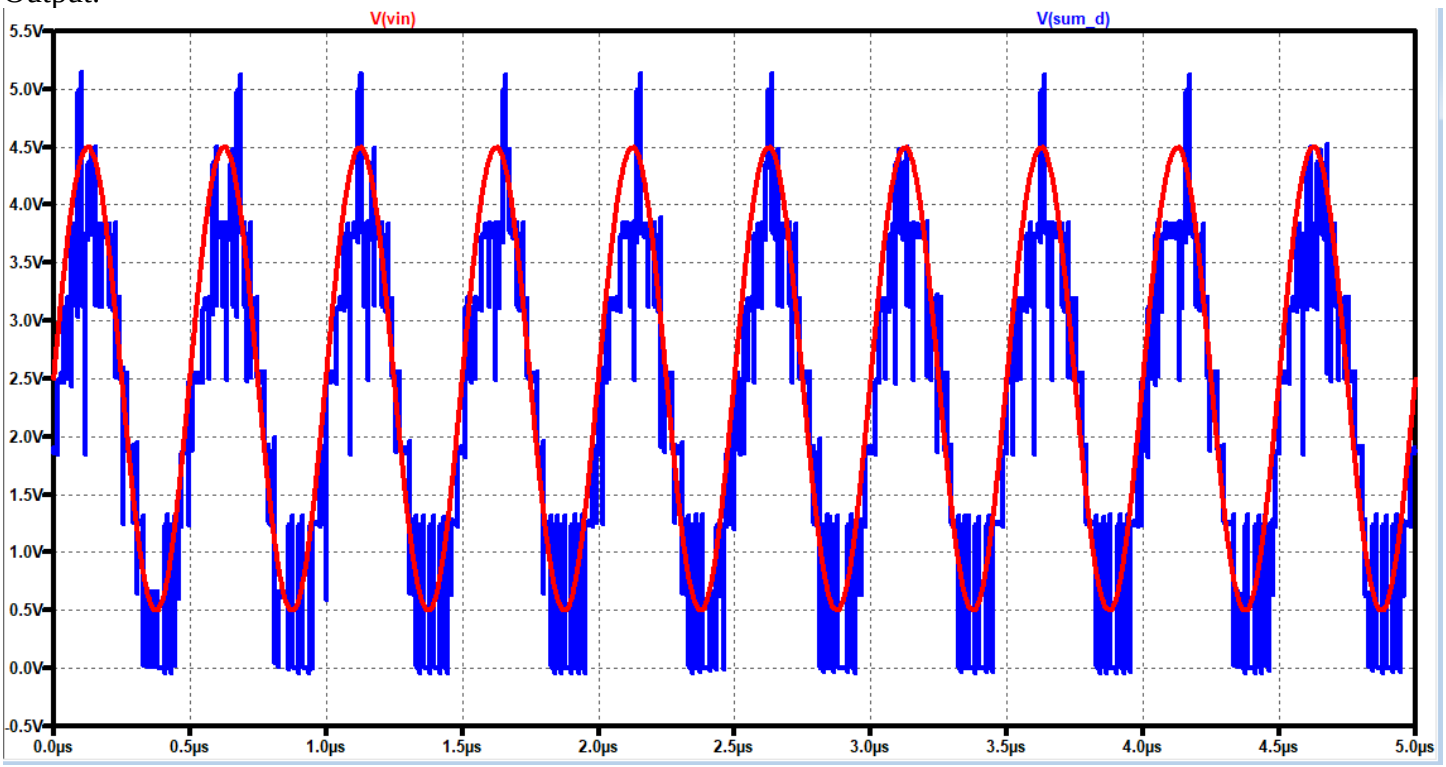
VDD	f_{in}	Amp.	R	R_1	C_1	C_2	VCO/Freq.	OSR	SNR (Parallel)	SNR (Series)	N_{EFF} (Parallel)	N_{EFF} (Series)
5V	2MHz	2V	6kΩ	6kΩ	8pF	1pF	2V/80MHz	64	29.99	33.12	4.69	5.21
5V	1MHz	2V	6kΩ	6kΩ	8pF	1pF	2V/80MHz	64	27.56	30.31	4.28	4.74
5V	1MHz	2V	6kΩ	6kΩ	8pF	1pF	2V/80MHz	64	27.56	28.89	4.30	4.50



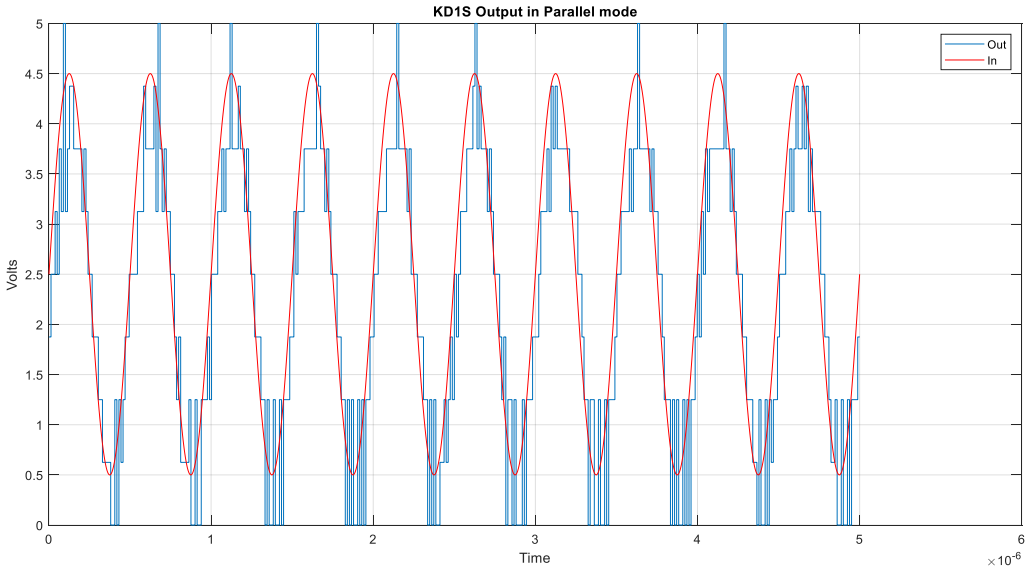
With this, all that was done was placing transmission gates at the feedback paths. Note that the timing diagram from the 1st order Transmission Gate design is still the same.



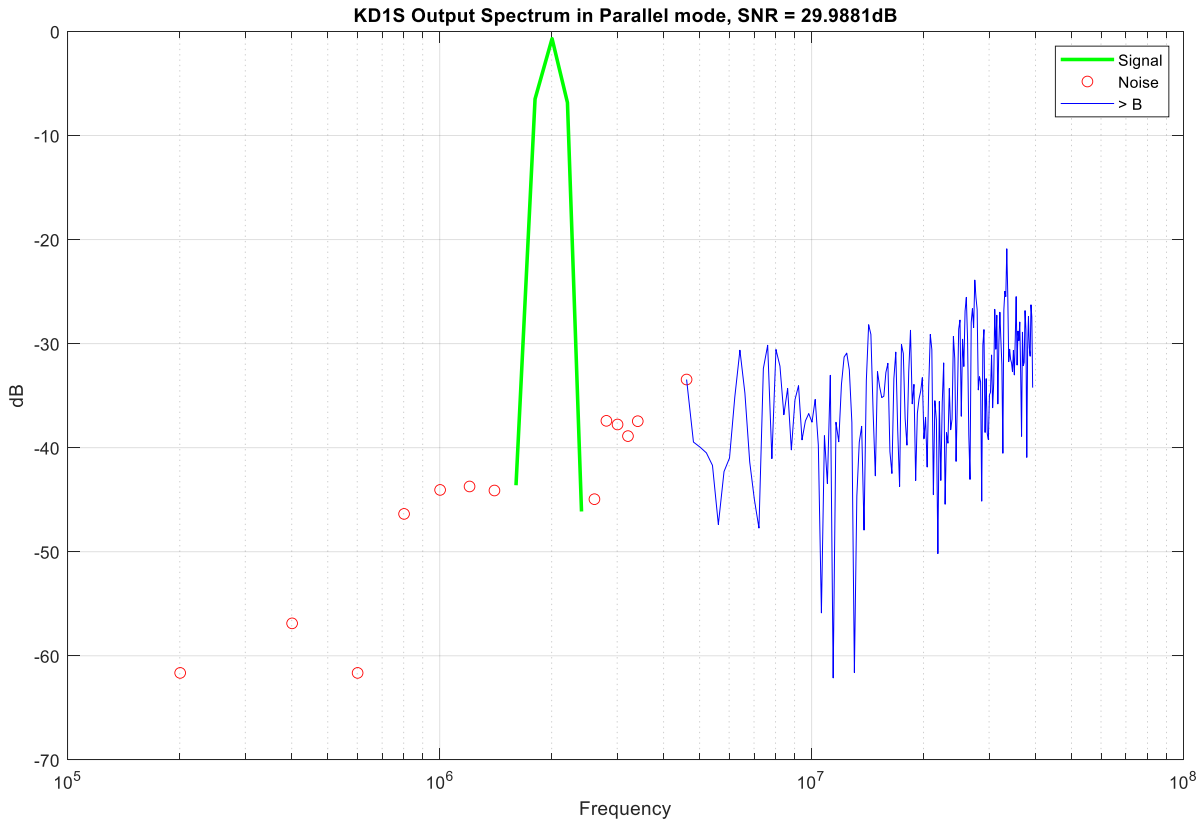
Output:



Successful transfer into MATLAB:



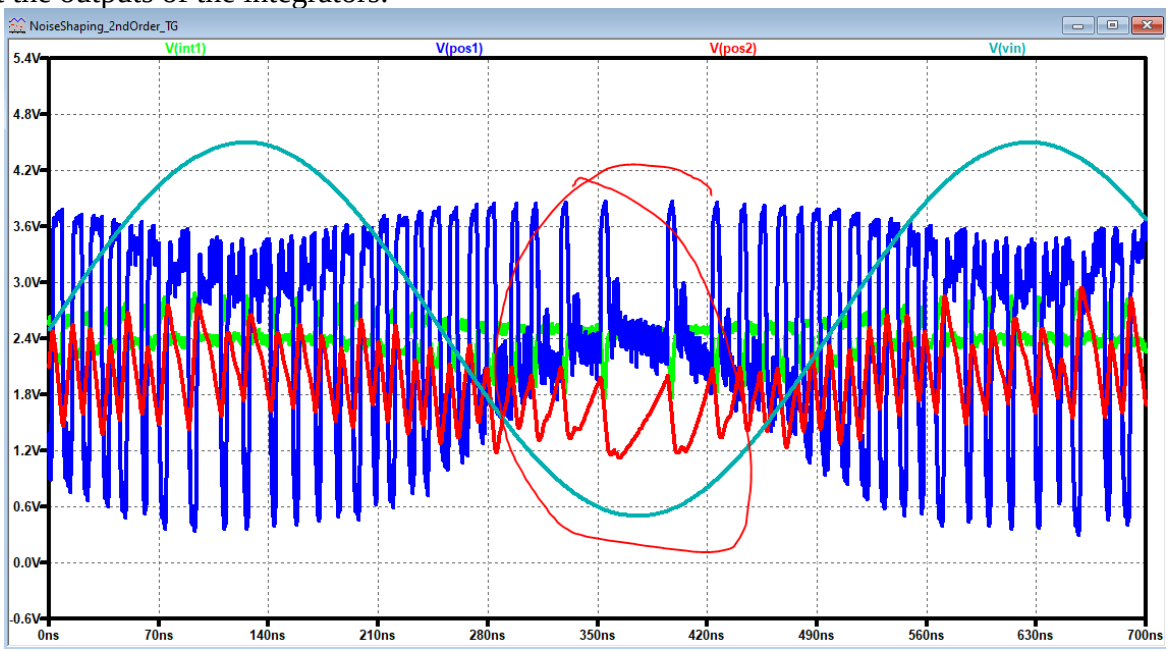
Viewing the SNR:



MATLAB output:

```
Kpath = 8 OSR = 64
For 1-bit output (serial mode) at Kpath*fs = 632 MHz
SNR = 33.12, Neff = 5.21, B = 4.94 MHz
For Kpath-bits output (parallel mode) at fs = 79 MHz
SNR = 29.99, Neff = 4.69, B = 4.92 MHz>>
```

Looking at the outputs of the integrators:



We do see areas where the output starts to saturate (bad)! The first integrator should have a larger capacitor to control this output swing more (at the cost of now having to improve the gain of the comparator).

Hand-Calculating SNR (2nd Order):

Recall the SNR equation for a 2nd order modulator:

$$SNR_{ideal} = 6.02N + 1.76 - 20\log\frac{(2\pi RC \cdot f_s)^2}{\sqrt{80}} + 50\log K$$

With our RC values of:

$$R = 6k\Omega, C = 2pF, K = 8, f_s = 80MHz$$

Our SNR is:

$$SNR_{ideal} = 6.024N + 1.76 - 20\log\frac{(2\pi(6k \cdot 2p) \cdot 80M)^2}{\sqrt{80}} + 50\log 8$$

$$SNR_{ideal} = 6.02N + 1.76 - 12.18 + 45, N = 1$$

$$\boxed{SNR_{ideal} = 40.59dB}$$

Comparing this to our Experimental Results for the 2nd order Large Resistor Topology:

VDD	***f*in**	**Amp.**	**R**	**R1**	**C1**	**C2**	**VCO/Freq.**	**OSR**	**SNR (Parallel)**	**SNR (Series)**	**NEFF (Parallel)**	**NEFF (Series)**
5V	2MHz	1.8V	6kΩ	6kΩ	5pF	2pF	2V/80MHz	64	38.95	38.76	6.17	6.14

We see here that we get close to our hand calculated SNR.

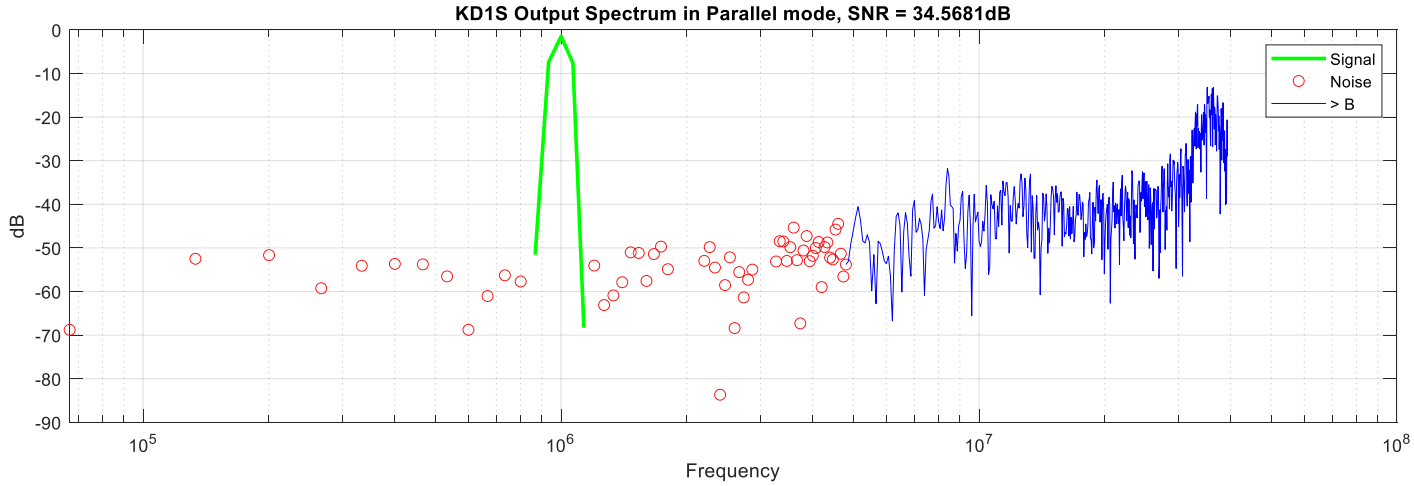
Things that could have been taken into account were that the components are non-ideal and will introduce their own distortion, causing the SNR to go down.

At this point, we are able to fully compare all of the designs.

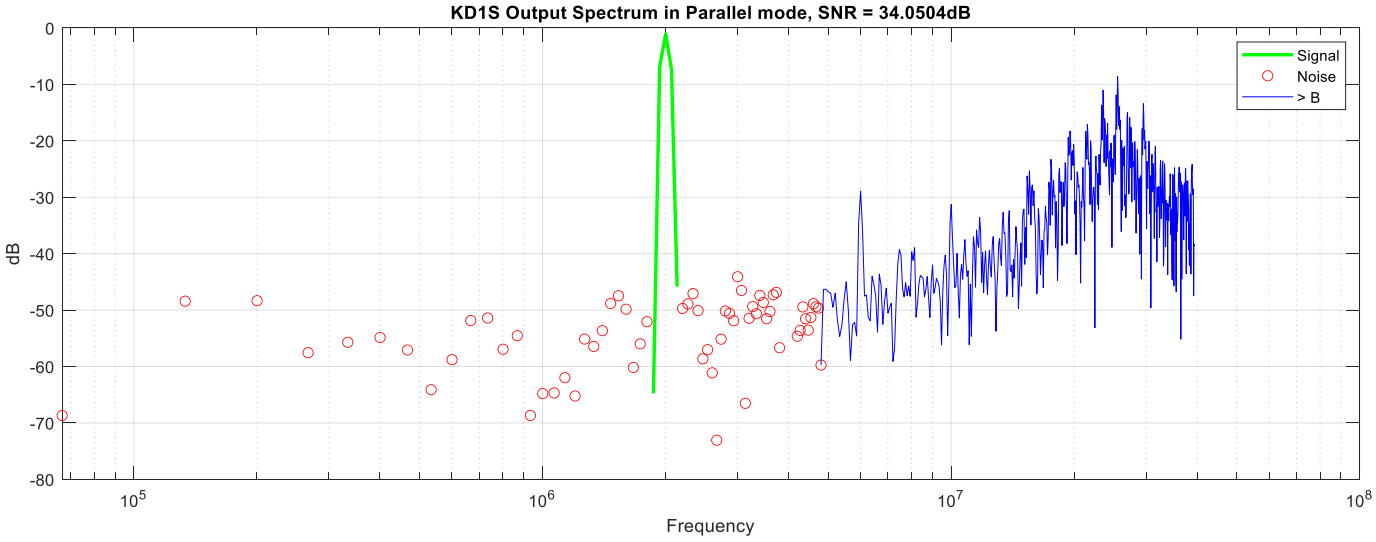
Doing a lengthy simulation of both modulators:

	1st Order w/Resistors	2nd Order w/Resistors	
$V_{In,Amplitude}$	1.8V	1.8V	
Sampling Frequency	80MHz	80MHz	
Kpath	8	8	
OSR	64	64	
SNR/Bits (Parallel)	34.57dB/5.45	34.05dB/5.36	
SNR/Bits (Serial)	34.92dB/5.50	35.08dB/5.53	
Bandwidth	4.92MHz	4.92MHz	

MATLAB FFT:
1st order w/Resistors:



2nd order w/Resistors:

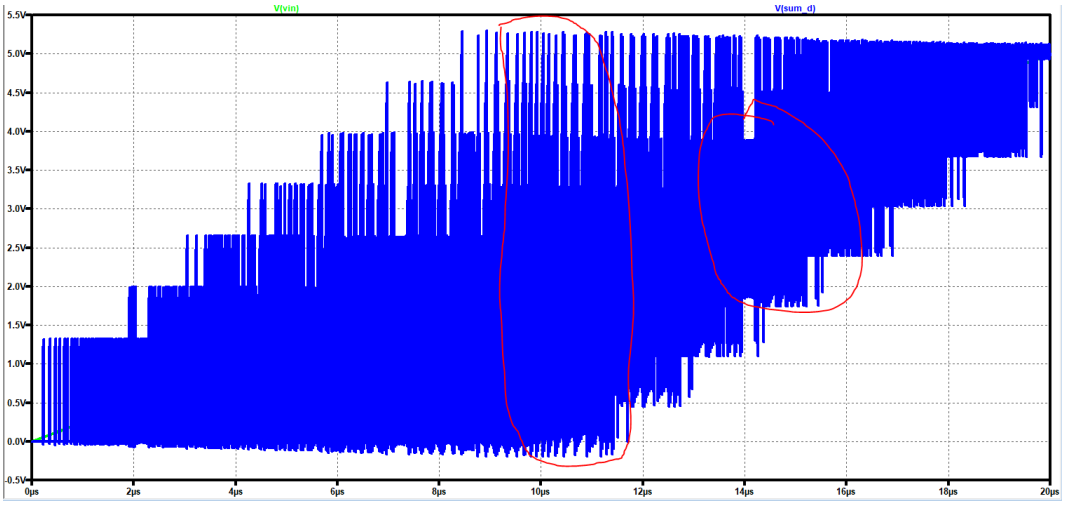


From above, we can see that the 2nd order noise shaping modulator has pushed all the noise to the higher frequencies.

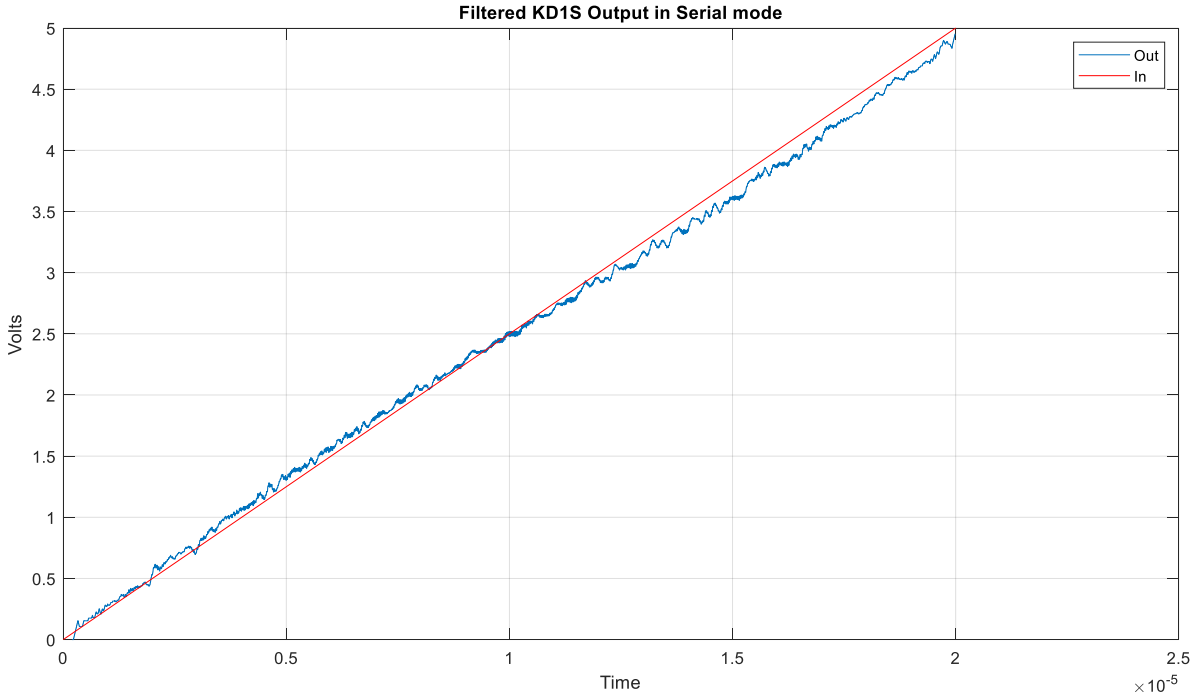
Linearity Tests:

Running a slow ramp and checking for deadzones:

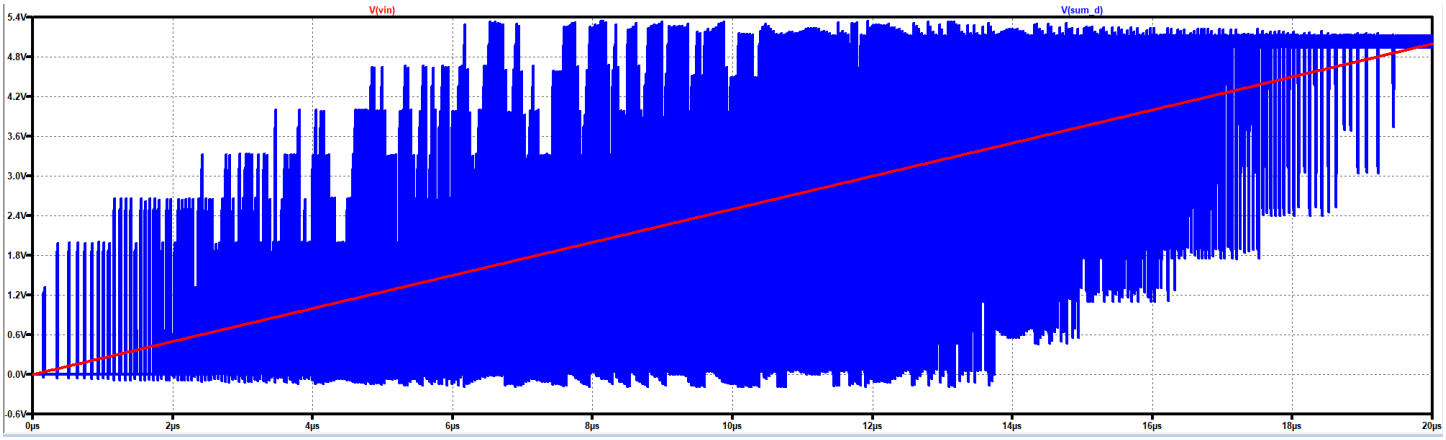
1st order W/Resistors:



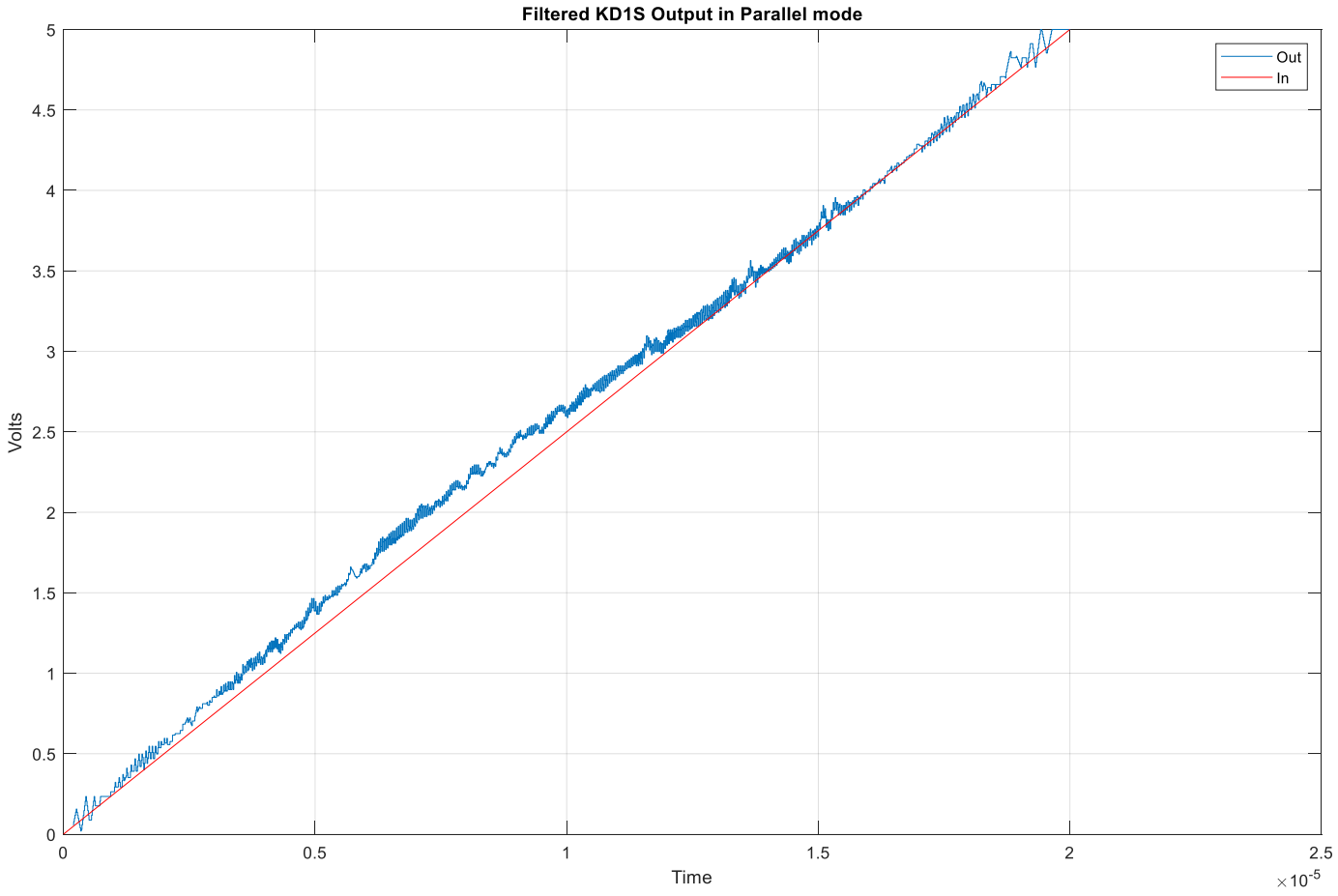
MATLAB's Filtered Output (to better see the dead-zones):



2nd Order W/Resistors:

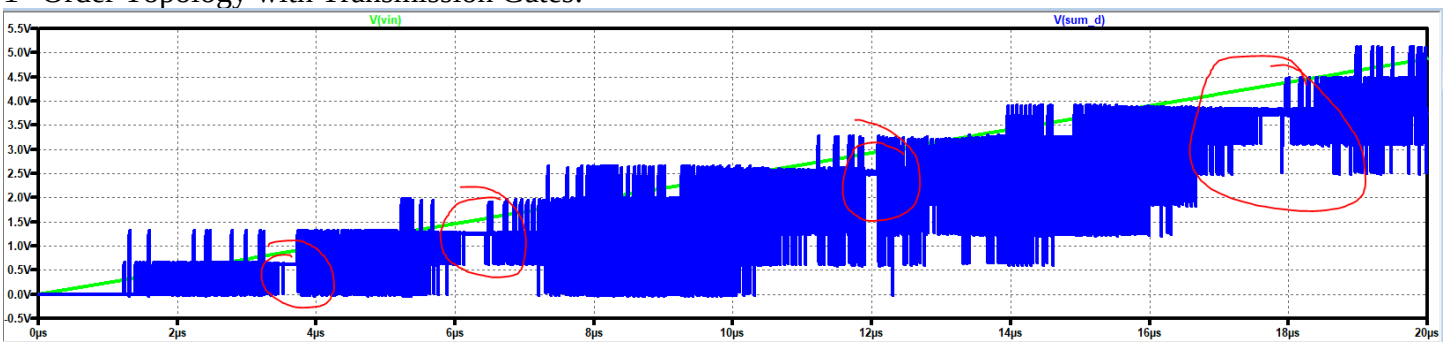


MATLAB Filtered Output:

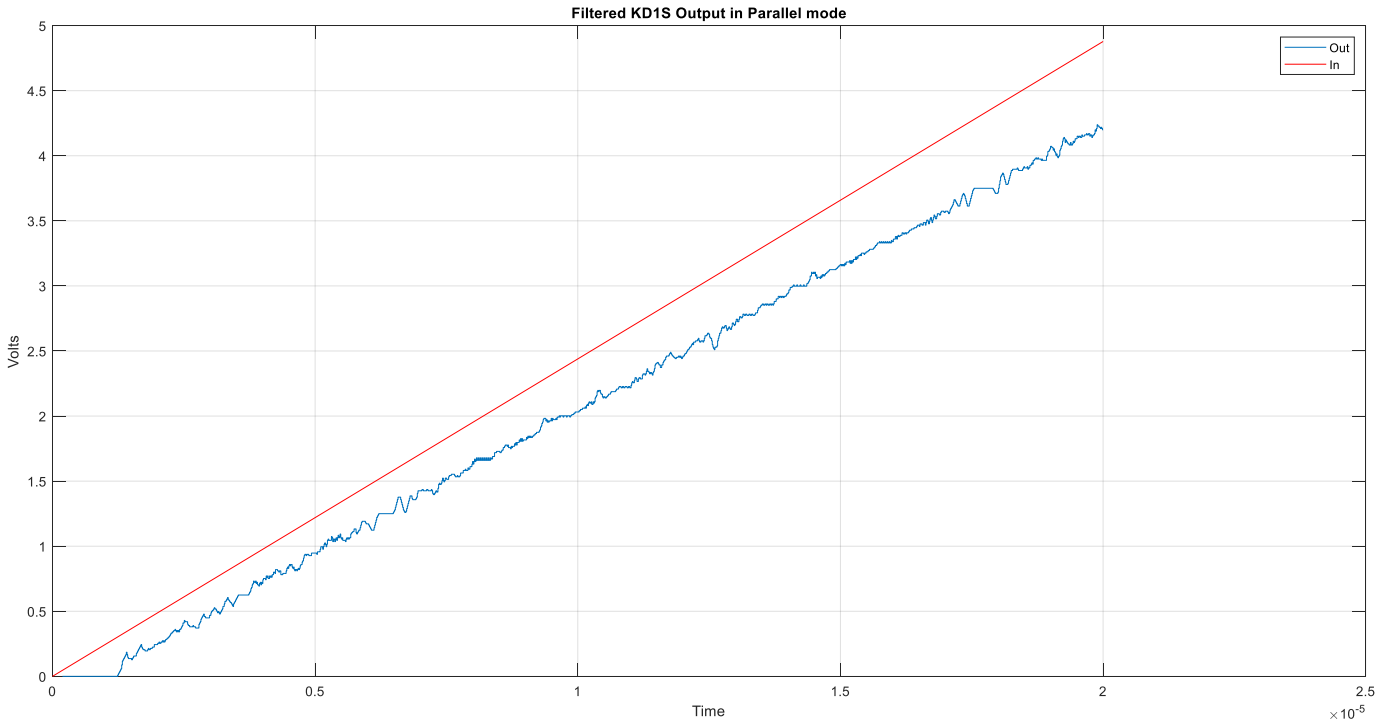


From the above 1st and 2nd order topologies for the large resistors, we see that the 2nd order has

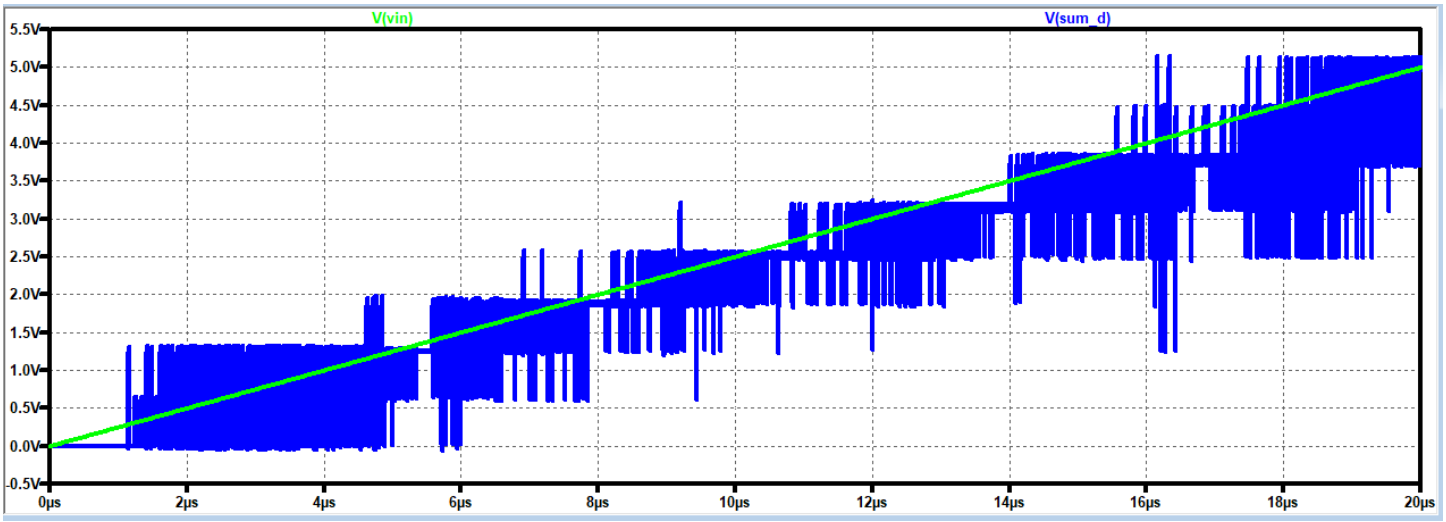
1st Order Topology with Transmission Gates:



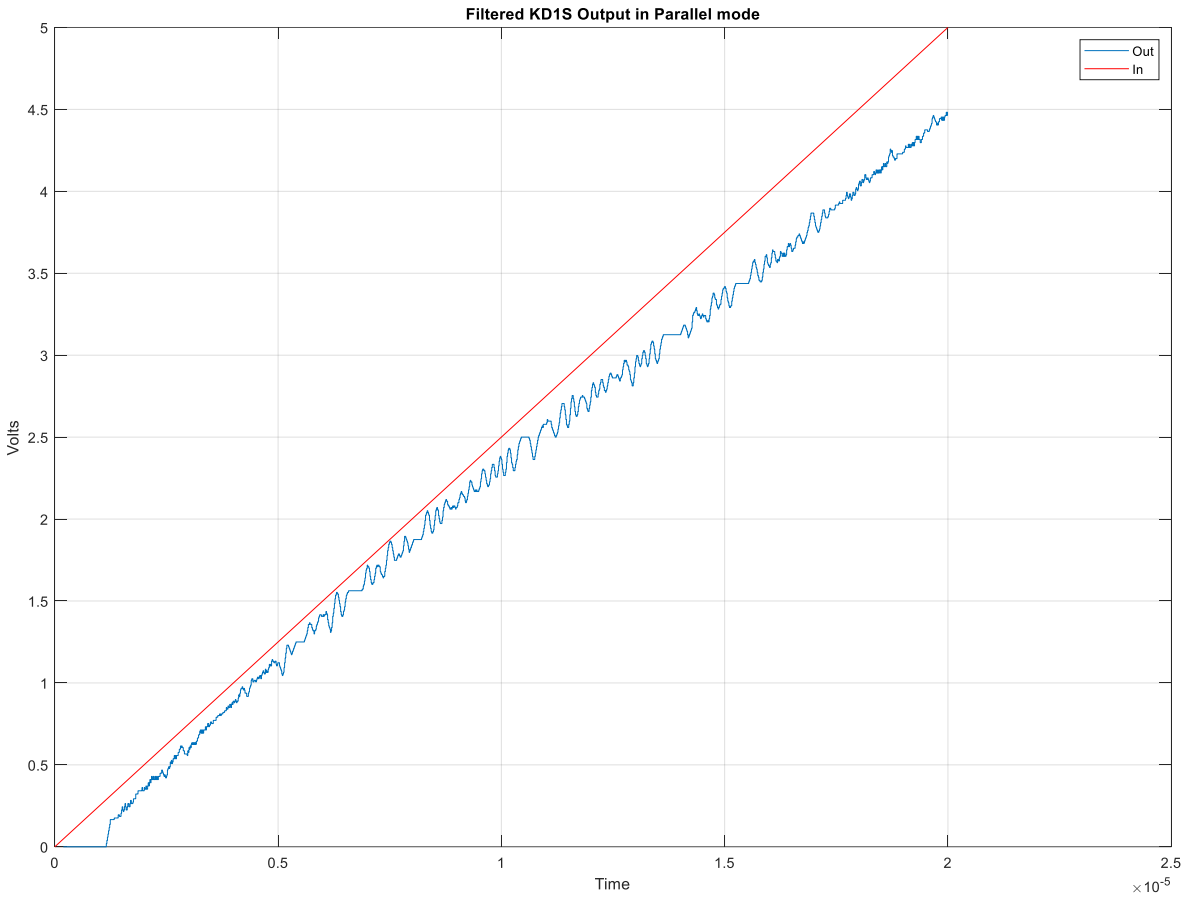
MATLAB Result:



2nd Order Topology with Transmission Gates



MATLAB Filtered Result:



In general, from the above 4 designs, we see that there are more dead-zones (areas with no activity) in the 1st order topologies than there are in the 2nd Order topologies. Note that for the 2nd order topology, the big weighing factor is the stability of the feedback loop.

Comparisons with Figure 9.32:

For each of the circuits, lets compare them to the circuit in the Mixed Signal Design book:

	1st Order Res	1st Order TG	2nd Order Res	2nd Order TG	Figure 9.32
Current Drawn (On Average)	10.24	11.64mA	18.17mA	18.19mA	17.28mA
VDD	5V	5V	5V	5V	5V
Power Used	51.2mW	58.2mW	90.85mW	90.95mW	86.4mW
Frequency	80MHz	80MHz	80MHz	80MHz	213MHz
SNR (Parallel)	35.76	39.37dB	38.95dB	27.67dB	37.84dB
SNR (Series)	38.54	35.72dB	38.76dB	28.89dB	38.92dB
Bits (Parallel)	5.64	6.24	6.17	4.3	5.99
Bits (Serial)	6.11	5.64	6.14	4.5	6.15
Bandwidth	4.93MHz	4.93MHz	4.93MHz	4.93MHz	13.4MHz

From the above, we can see that with the 2nd Order topologies, we have more current drawn due to the integrators. The other thing that is interesting is that the Resistor topologies are doing exceptionally well and give higher SNR values, however, at the cost of huge layout space.

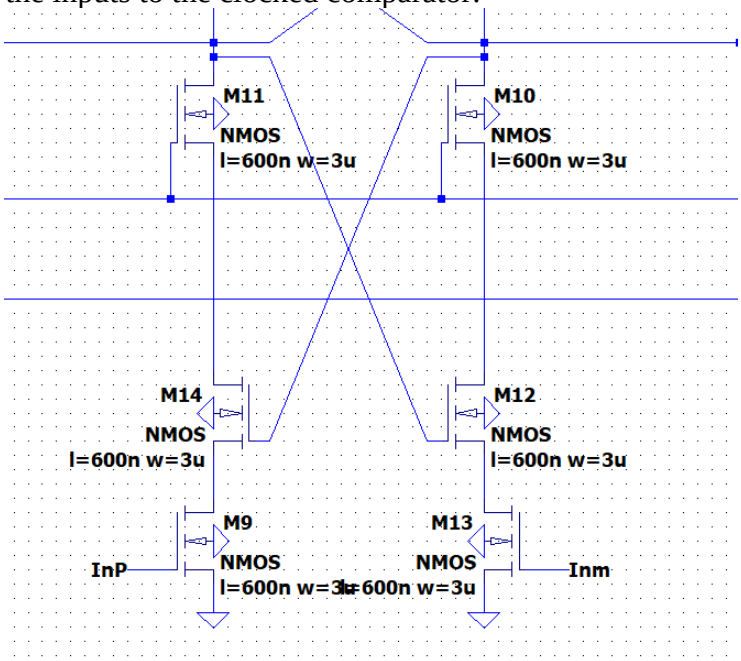
Regarding the power consumption of the 1st order topologies, there is a higher chance of stability for the integrators and therefore will not make the integrator work too hard.

Conclusions and areas to improve

For the above topologies, there are a few assumptions that were made to make our designs practically work.

One of the biggest things to investigate is the design of a better clocked comparator.

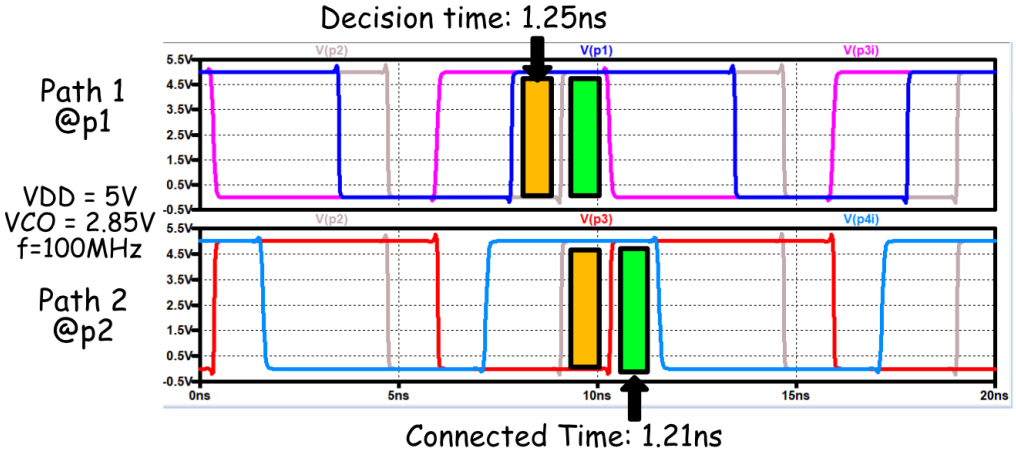
Recall the primary stage of the inputs to the clocked comparator:



In analog design, a preamplifier stage (a self-biased diff amp) would be able to increase the gain and sensitivity in this circuit, which will be able to result in having to design a more stable and fast integrator.

The other investigation that can clearly be looked into is the timing of how much charge is injected into a path before opening.

Recall the timing diagrams for the clocks:



The decision time for the circuits was a narrow window of around $\frac{f_s}{8}$. This is the ideal time that we would want so that our modulator is accurate, however, if the sampling frequency is too high, then there is a chance that the clocked comparator will not make a choice and we end up with less signal feeding into the integrator.