EE-523 ADVANCED ANALOG INTEGRATED CIRCUIT DESIGN TP-2025

PRACTICAL EXERCISE SESSION No. 4

Note: For students connecting virtually, the details of the Zoom Meeting for this TP session are given below

Topic: Advanced Analog IC Design Zoom Meeting

Time: April 11th, 2025 01:00 PM Amsterdam, Berlin, Rome, Stockholm, Vienna

Join Zoom Meeting:

https://epfl.zoom.us/j/66887254608?pwd=MBLOAjyTb7PaR02tNUeDaHDb1q5WID.1

Meeting ID: 668 8725 4608

Passcode: 191600

Objectives of this Practical Exercise Session

1. Design and simulate a non-overlapping clock generator:

    a) Include the ahdlLib library to use logic gates in virtuoso.
    b) Following the design procedures, design a non-overlapping clock generator.
    c) Perform transient simulation to verify the functionality of the clock generator.

2. Design and simulate a switched-capacitor amplifier:

    a) Following the design procedures, create a symbol for the OTA from TP2.
    b) Design a switched-capacitor amplifier applying the OTA and the non-overlapping clock generator.
    c) Perform transient simulation to check the performance of the switched-capacitor amplifier.
    d) Simulate the switch linearity.
    e) Introduction to the bootstrapped switch.

1. Design and Simulate a Non-Overlapping Clock Generator

Design Specification

- Non-Overlapping Time > 2ns
- Operating Frequency : 10MHz

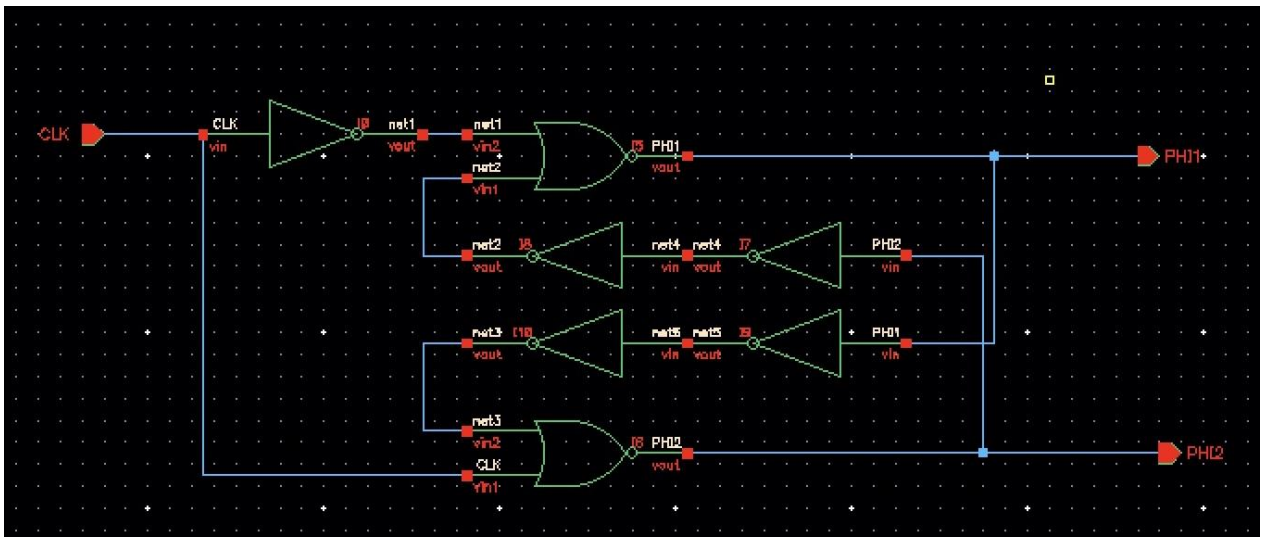
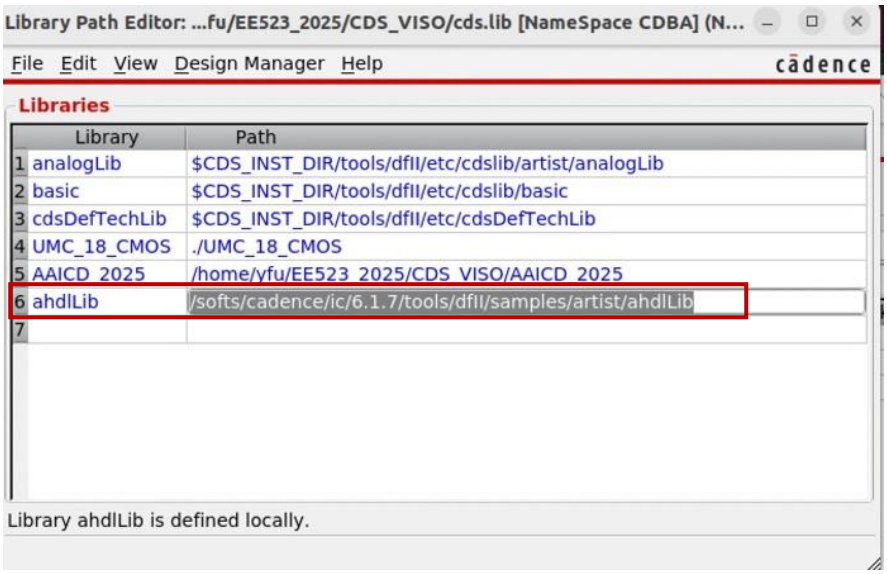


Fig. 1: Schematic of the Non-Overlapping Clock Generator

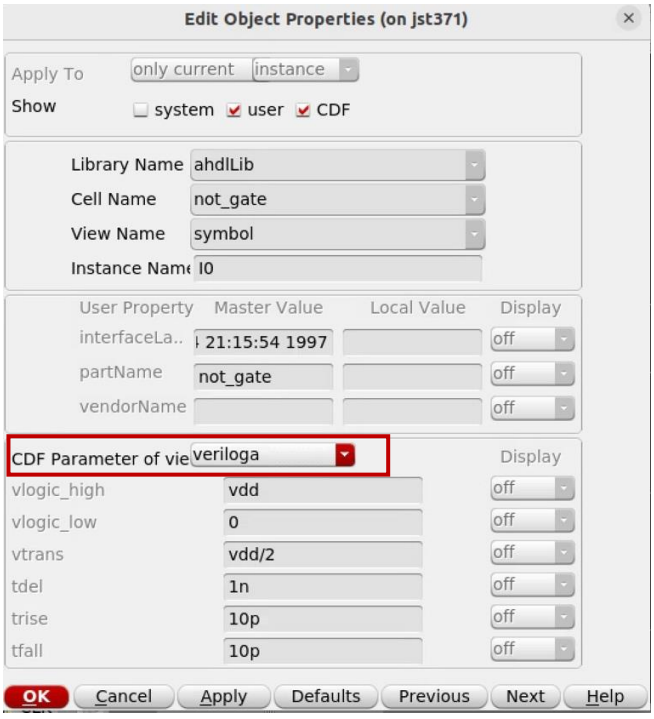
The non-overlapping clock generator is a very important block for switched-capacitor circuit design. It helps avoid issues such as unintended charge sharing and short-circuit during the switching between different phases. The circuit schematic is shown in Fig. 1. The two inputs to the NOR gate in this circuit can either be 11 or 00. In both cases, due to the cross-coupled configuration, the outputs will stabilize to either 11 or 00, depending on the previous state and the input transitions. This feedback mechanism ensures that the outputs are mutually exclusive and self-recovering, preventing both PHI1 and PHI2 from being high (1) or low (0) at the same time. As a result, PHI1 and PHI2 always maintain complementary or non-overlapping states, which is essential for proper clock phase alternation in dynamic logic circuits. The non-lapping time is mostly dependent on the delay of the inverters in the cross-coupling path.

Building the Schematic

To build the schematic of the non-overlapping clock generator, we need to refer to the ahdlLib library, which provides models for different digital components. Here we do not use the digital standard cells to reduce the difficulty in getting sufficient delay. Besides, it is a reasonable to start with the VerilogA model than directly working on the standard cell to check the functionality of the circuit. To include the VerilogA library, you could first go to the library manager and click **Edit>Library Path** to include the ahdlLib: '/softs/cadence/ic/6.1.7/tools/dfll/samples/artist/ahdlLib' as shown below. Remember to choose 'save the changes' when closing the library path window.



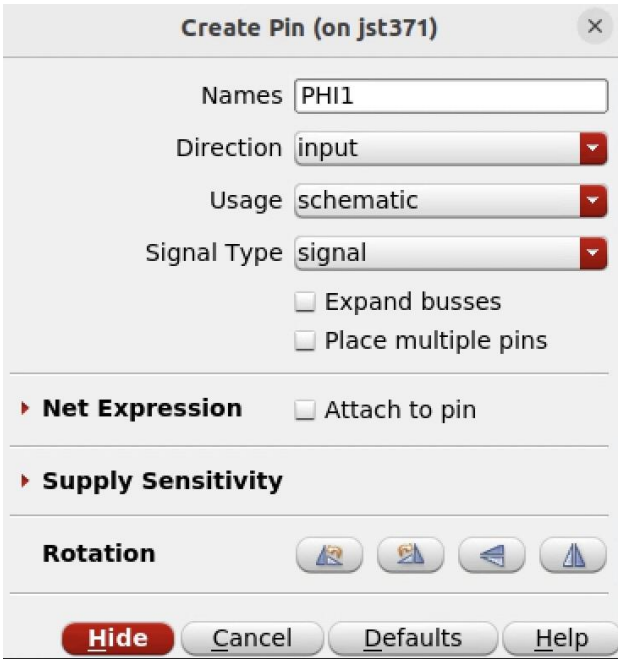
Instantiate the ‘not_gate’ and ‘nor_gate’ from the ahdlLib, and construct the schematic as shown in Fig. 1. For all the logic gate, you could set the parameters to the value shown in the following table. In the Edit Object Properties window, the CDF Parameter of view should be changed to ‘veriloga’ to set up these parameters.



vlogic_high	vlogic_low	vtrans	tdel	trise	tfall
vdd	0	0.5*vdd	1n	10p	10p

Create a Symbol for the Schematic

For hierarchical design, a symbol for the schematic is required. Before creating a symbol, we need to define the pins. Press ‘P’ to create pins for the input, output and supply nodes. Here we only have input nodes and output nodes. Add the input pin ‘CLK’ and the output pins ‘PHI1’ and ‘PHI2’ (as shown in Fig. 1).



After adding the pins, click **Create>Cellview>From Cellview** and press OK to generate the symbol. In the library manager the you will find the symbol view.

You could edit the symbol to make it look more representative of the circuit functionality. Here, we added some pulse drawing to show that this block is a non-overlapping clock generator, as shown in Fig 2.

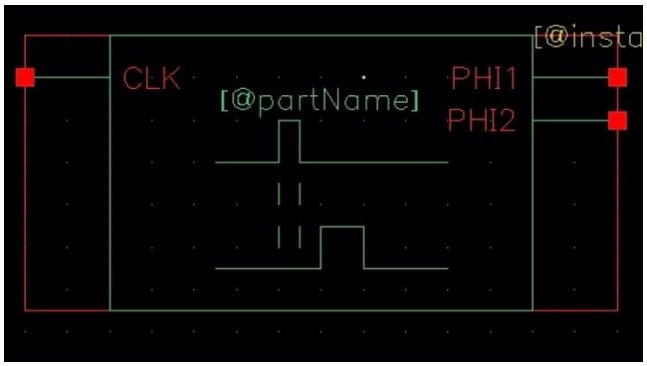
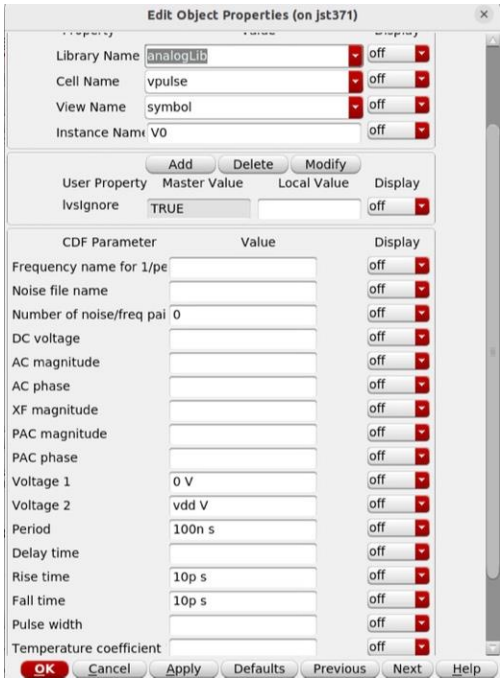


Fig. 2: Symbol of the Non-Overlapping Clock Generator

Create a Testbench for the Non-Overlapping Clock Generator

In order to test the Non-Overlapping Clock Generator, we need to create a testbench. We could create a new schematic called ‘tb_CLK_GEN’ and instantiate the symbol of the non-overlapping clock generator. To create the input clock for the non-overlapping clock generator, you could instantiate the ‘vpulse’ block from the ‘analogLib’ and setup as below. The duty cycle by default is 50%, and you could setup the duty cycle by editing the pulse width. Here PHI1 and PHI2 are not connected to anything. As shown in Fig. 3, we could instantiate the ‘noConn’ block from the ‘basic’ library and connect to them to avoid warnings during check and save of the schematic.



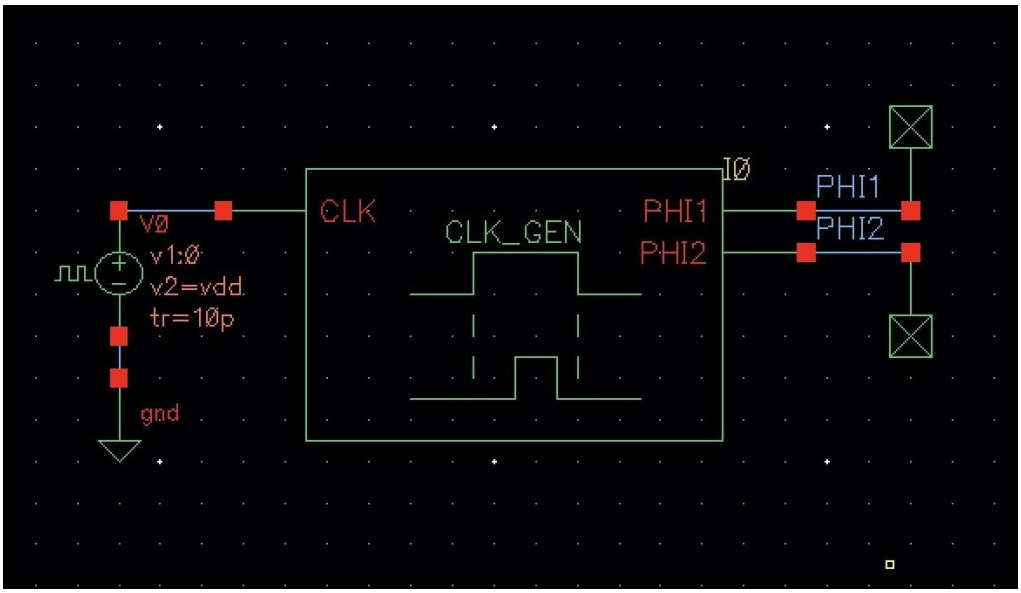
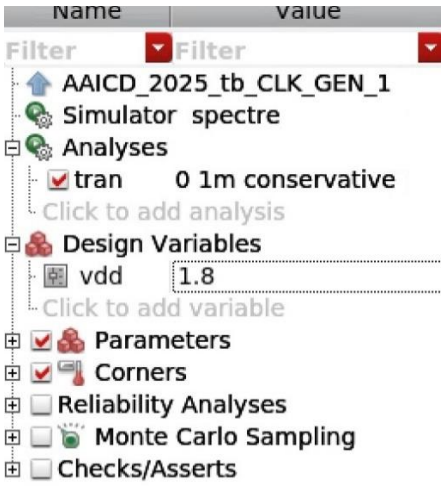


Fig. 3: Schematic of the Non-Overlapping Clock Generator Testbench

Launch the ADE Explorer and setup transient analysis to check the functionality of the block. The accuracy of the simulation could be set as ‘conservative’ and the simulation time could be set as 1ms. The vdd could be set as 1.8V.



Run the transient simulation and plot the non-overlapping clock PHI1 and PHI2.

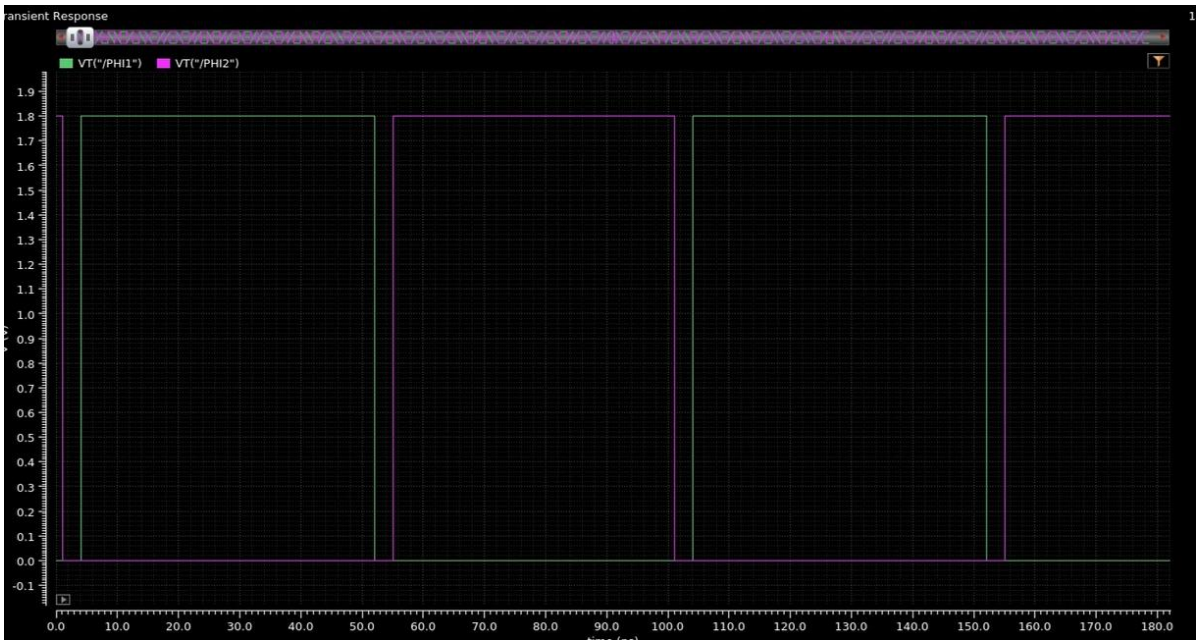


Fig. 4: Transient Simulation of Non-Overlapping Clock Generator

Verify the non-overlapping time of PHI1 and PHI2.

2. Design and Simulate a Switched-Capacitor Amplifier

Design Specification

- Sampling Frequency 10MHz
- Gain of 2

For the OTA of the switch capacitor amplifier, we will reuse the OTA designed in TP2. The parameters of the OTA are listed below. The schematic in TP2 contains both the stimulus (voltage sources for input, power, etc.) and the OTA itself.

I_{BIAS}	L_{TAIL}	W_{TAIL}	F_{TAIL}	L_n	W_n	F_n	L_p	W_p	F_p	V_{b2}	V_{b1}
19 μA	720 nm	1 μm	2	720 nm	1 μm	7	360 nm	1 μm	12	0.8	1.2

To use the OTA in the hierarchical design, we remove the stimulus and put pins wherever necessary, similar to what we did for the non-overlapping clock generator in the first part of this TP, as shown in Fig. 5.

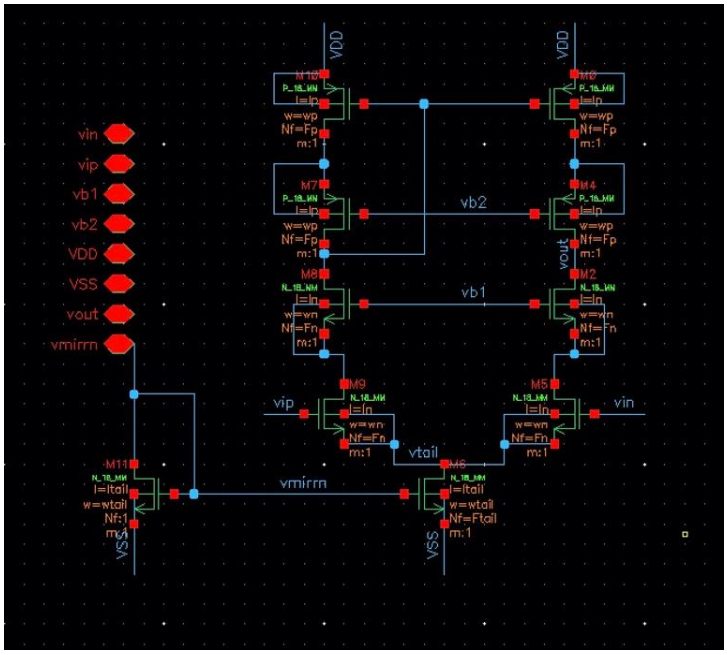


Fig. 5: Schematic of the OTA with Pins

After editing the OTA schematic, you can create a symbol similar to how we did it for the non-overlapping clock generator, shown in the first part of this TP. As shown in Fig. 6, it is recommended to put the inputs on the left, output on the right, bias voltage, bias current and supply voltage on top, and the ground connection on the bottom.

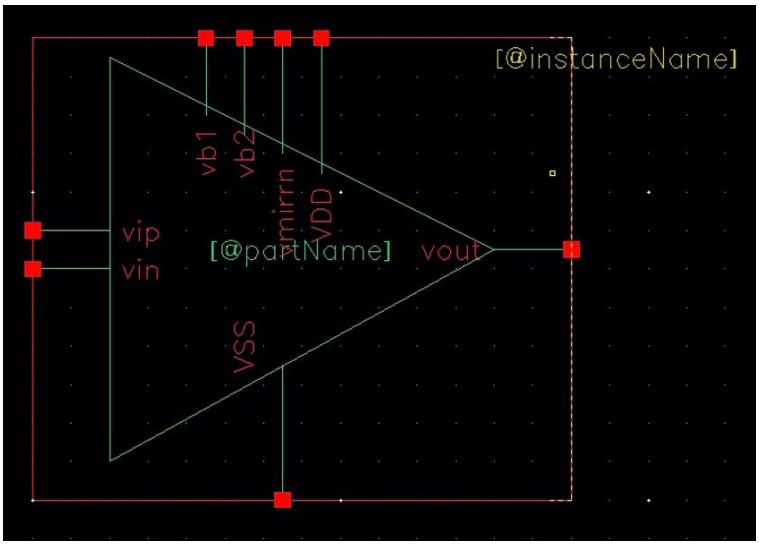
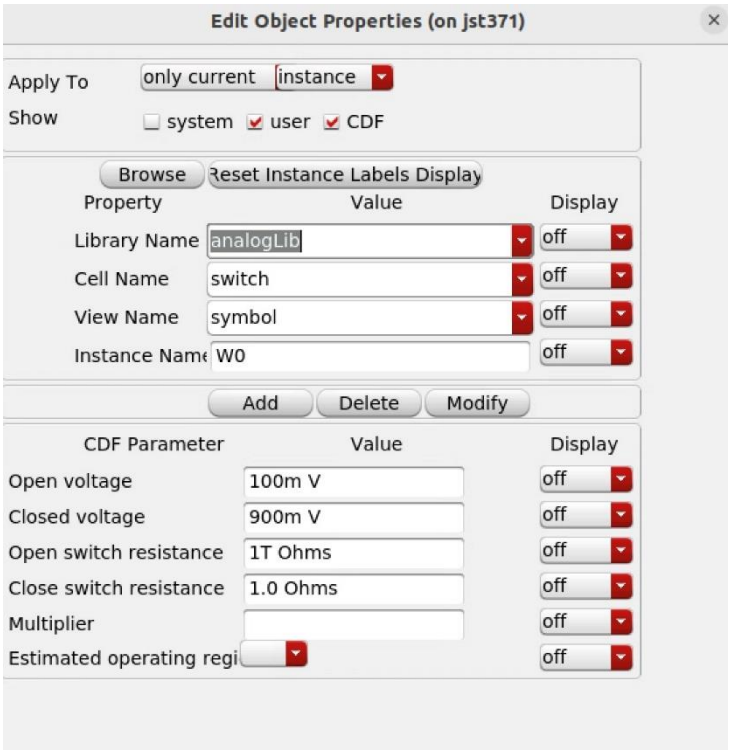


Fig. 6: Symbol of the OTA

Now, we are ready to build the switched-capacitor amplifier. For the switches in the switched-capacitor amplifier, we use ideal switch (from analogLib). You could setup the switch from analogLib as shown below. The open voltage could be set to 100mV and the closed voltage could be set to 900mV. The open switch resistance and the closed switch resistance could be the default value. The switch is a four-terminal device, two terminals are for the input and output of the switch and the other two terminals are for the control voltage. The on and off of the switch is controlled by the voltage difference between the control terminals.



The schematic of the switched-capacitor amplifier is shown in Fig 7. During the PHI1 phase, the input capacitance (C0 in Fig. 7) tracks the input. During the PHI2 phase, the charge on the input capacitor is transferred to the feedback capacitor (C1). The input voltage could be attenuated or amplified based on the capacitor ratio between the input capacitor and the feedback capacitor. We choose the input capacitor as 2pF and for the gain of 2, the feedback capacitor is set to 1pF. The non-overlapping clock generator and the OTA are instantiated and integrated in the schematic. The stimulus/excitation for the OTA - vb1, vb2, ibias, etc. and the non-overlapping clock generator is the same as the TP2 setup and the previous part of this TP respectively. Here, the common-mode voltage vcm is set to 900mV and the vdc of the vin is set to 900mV since it is the recommended common mode input for the OTA in TP2.

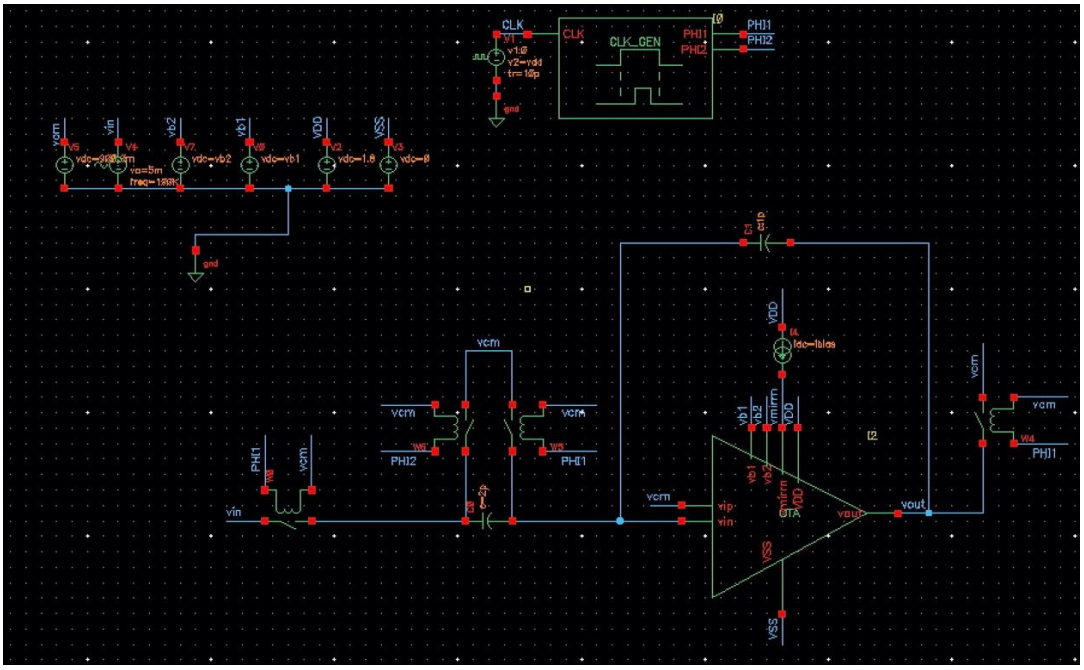


Fig. 7: Schematic of the Switched-Capacitor Amplifier Testbench

After finishing the schematic, you could set up the transient simulation similar to the TP2 to simulate the switched-capacitor amplifier.

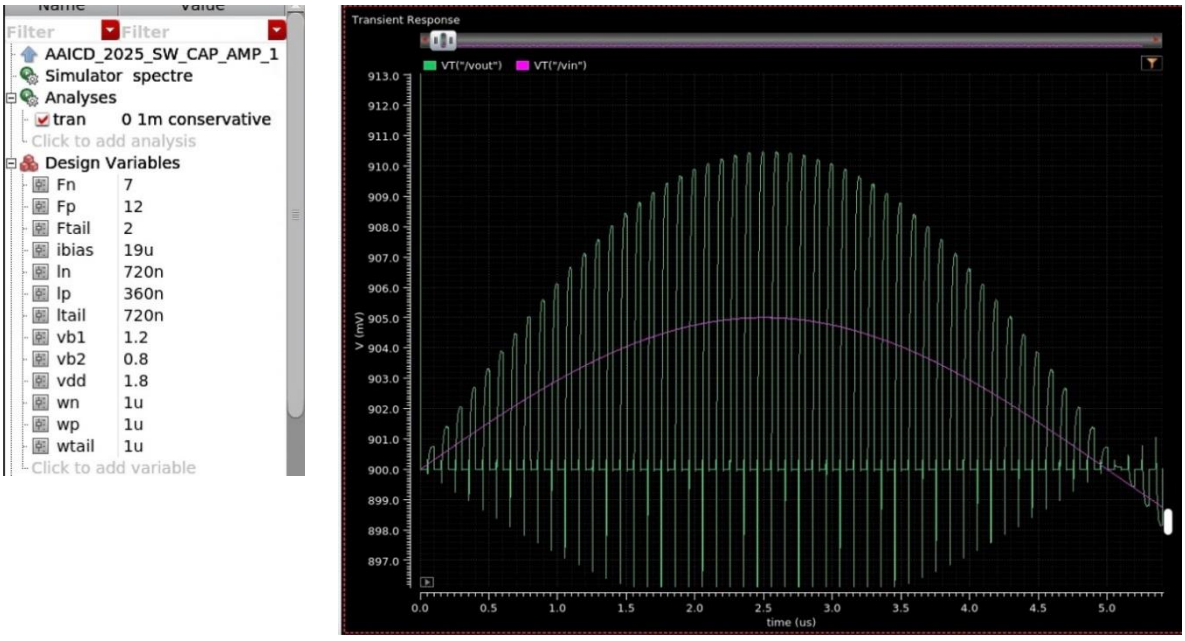


Fig. 8: Simulation Setup and Transient Waveforms of the Switched-Capacitor Amplifier

Now, you could try to replace the ideal switches with the real CMOS switches, as shown in Fig. 9. You could first add ideal inverters from the ahdlLib for PHI1 and PHI2 to generate the inverse of them. This time you don't need to edit the parameter of the inverters since PHI1_B and PHI2_B are the complete inverses of already generated non-overlapping clocks PHI1 and PHI2. For all the NMOS transistors the W is set to 240nm and L is set to 180nm. For all the PMOS transistors the W is set to 480nm and L is set to 180nm. The bodies of all the transistors are connected to the source.

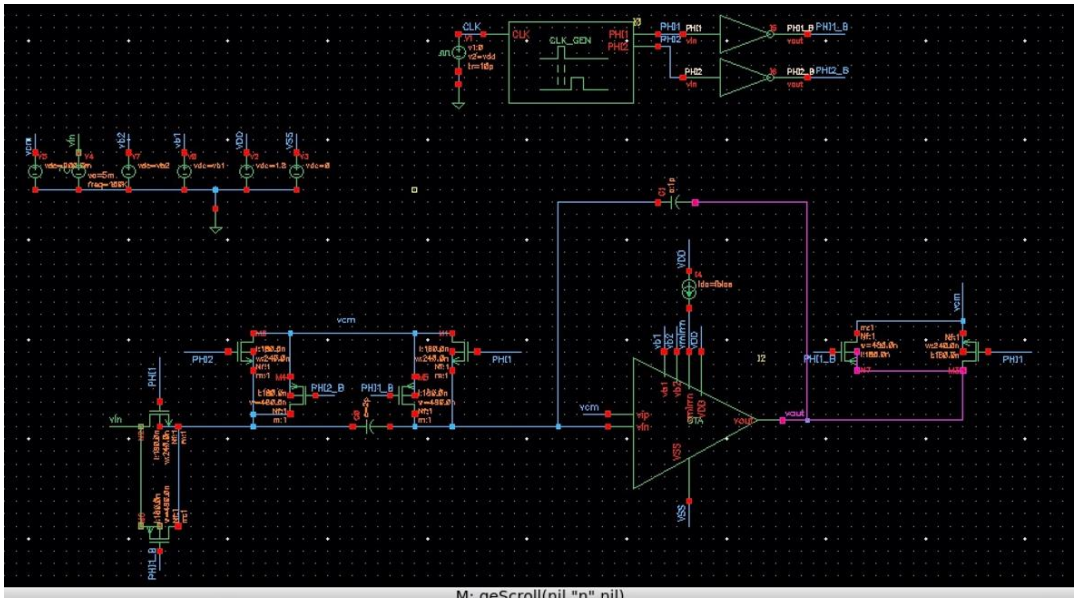
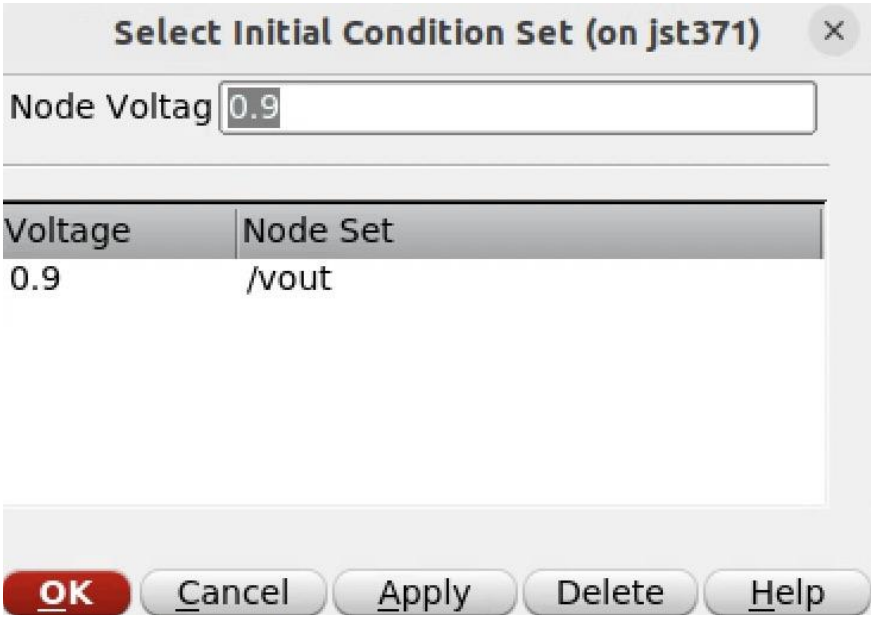


Fig. 9: Schematic of the Switched-Capacitor Amplifier with CMOS switches

Before you run the simulation,click on **Simulation>Convergence aids>Initial condition.** Enter 0.9 as the 'Node Voltage' and then click on the 'vout' net in the schematic to set its initial value to 0.9V. This makes the plot more viewable in this case.



Run the transient simulation and check the result. As shown in the Fig 10, comparing to the ideal switch case, the cmos switch case includes the effects of capacitor coupling.

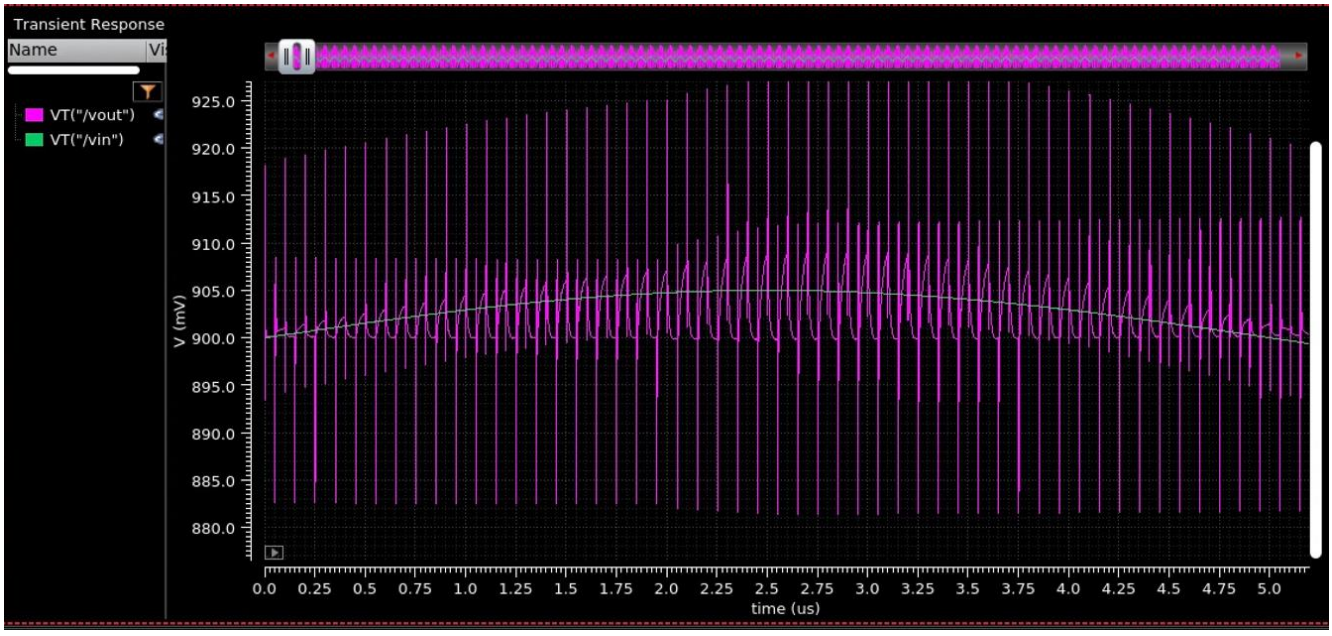
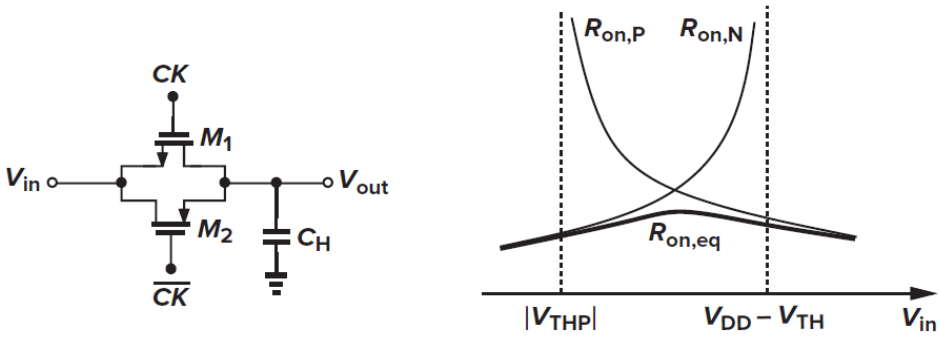


Fig. 10: Simulation result of the Switched-Capacitor Amplifier with CMOS switches

Switch Selection – Linearity Considerations

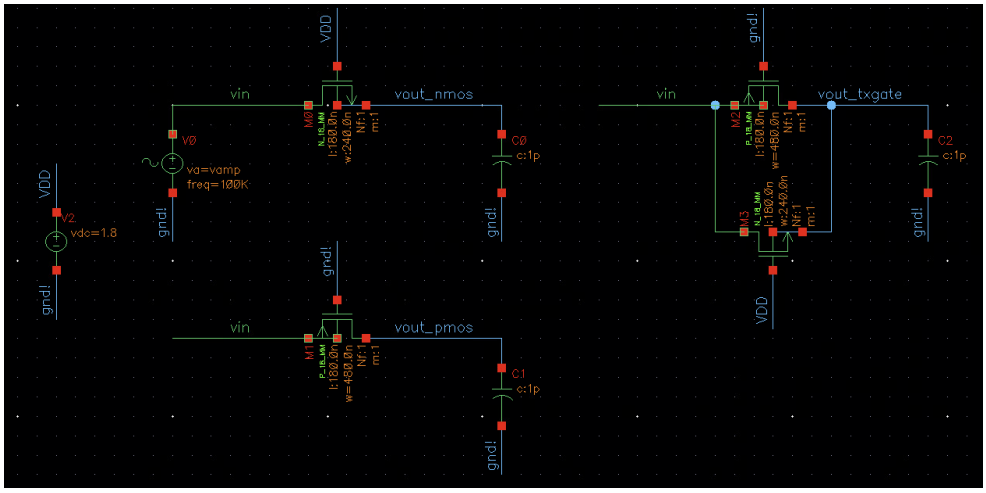


At this juncture, we take a small detour to simulate the linearity of different switches, namely, NMOS, PMOS, and the Transmission Gate (TX Gate). As shown in the figure above, we expect the TX Gate to be the most linear switch as its ON resistance is almost flat across the input range. However, we note that the NMOS is also as linear as the TX Gate when the input is low and the PMOS as linear as the TX Gate when the input is high. However, it seems like the TX Gate is always the safer option.

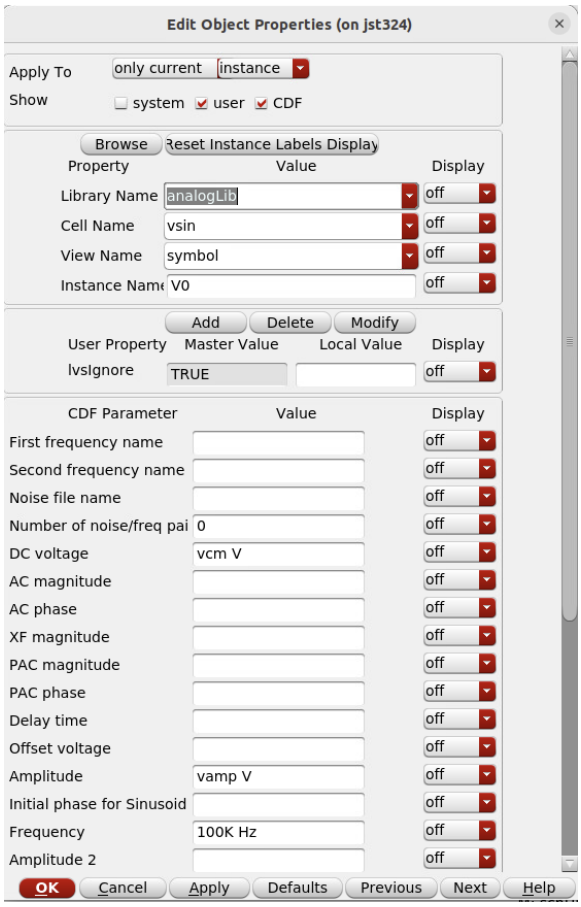
In the following experiment, we will try to simulate and see the linearity as a **function of the input common-mode and signal amplitude.** We will not perform the sampling operation, as our goal is to solely analyze the linearity of the switches and the sampling operation will complicate the frequency spectrum and take away the intuition. However, with this simplification, we note that we will miss some aspects like charge injection, clock feedthrough, etc. that affect systems with sampling.

The interested reader is encouraged to extend the following simulation for the case with sampling.

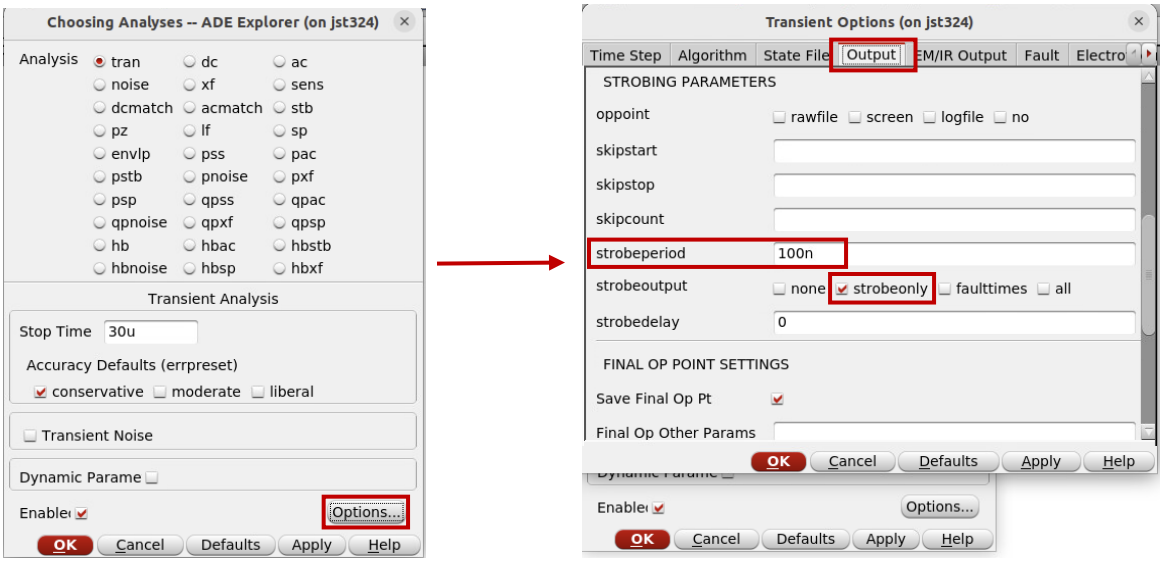
Create a new cellview named “TB_SW_LINEARITY” and construct the schematic as shown below. The NMOS switch is of minimum size (W_n = 240n and L_n = 180n), the PMOS is twice the minimum width (W_p = 480n and L_p = 180n). The TX Gate uses these two switches in parallel. We note that the gate for these switches is connected to VDD (for NMOS) and gnd! (for PMOS), so that they are in their ON states.



The properties for the vsin source connected to vin are shown in the figure below.



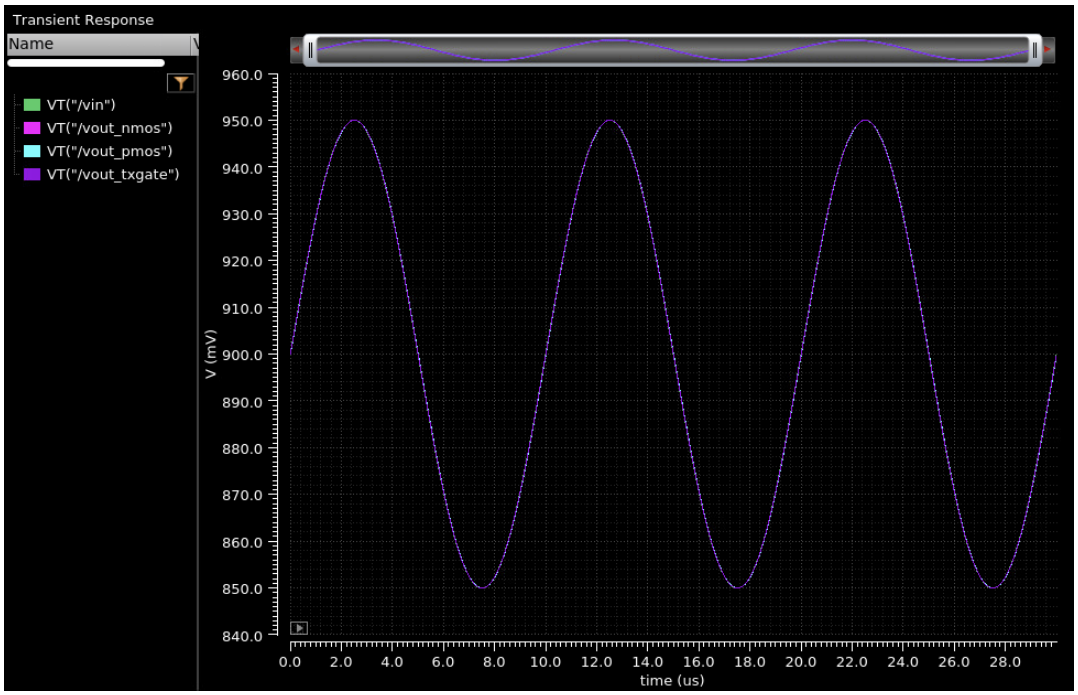
Launch ADE-Explorer to run the simulation. **vcm is set to 900m and vamp to 50m for this simulation**. We will run the transient simulation to check the linearity of the output signals (there are other analysis types to check the linearity, but we will stick to transient as it is the most accurate one). Setup the transient simulation as shown below.



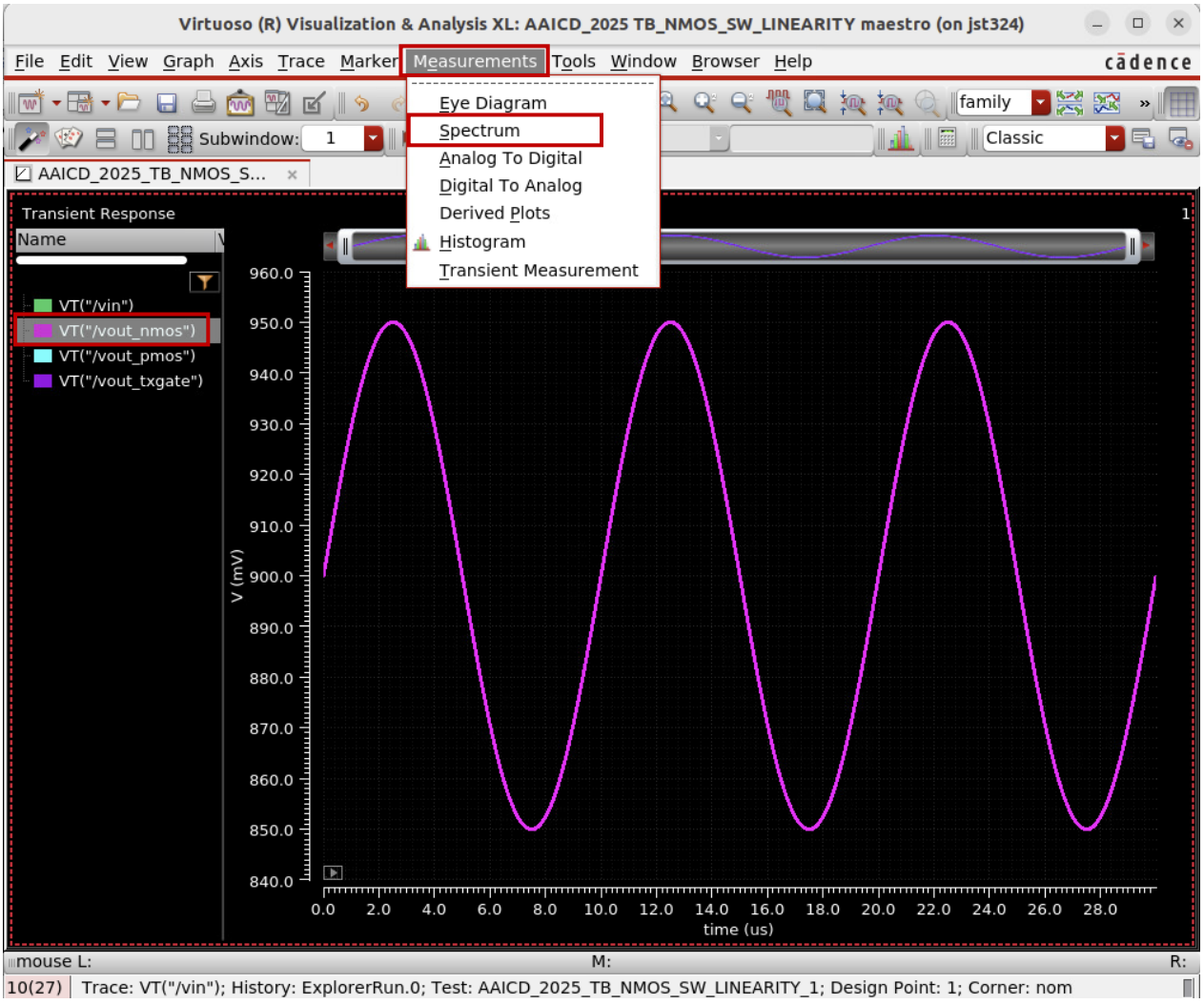
After setting the ‘Stop Time’ and Accuracy setting, click on Options. Now, under the ‘Output’ tab of the Transient Options form, set the ‘strobeperiod’ to 100n and ‘strobeoutput’ to strobeonly. This tells the simulator to run the computations for the transient simulation every 100n and to store only these points in the final results.

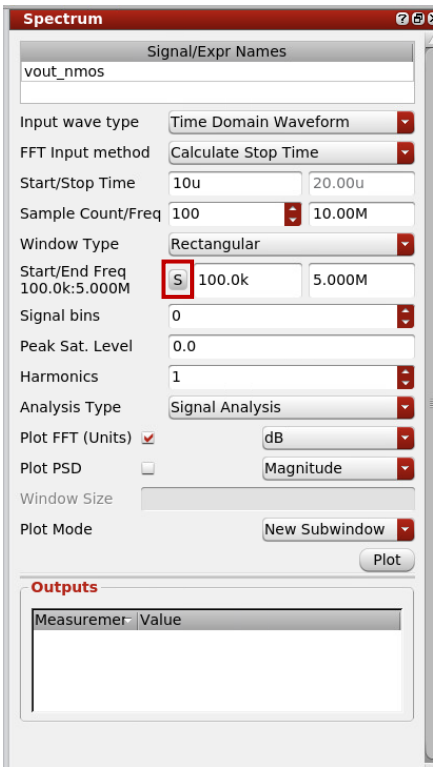
If this is not clear or you want to know more, please grab hold of one of the TAs!

Now, run the simulation and plot the input waveform (‘vin’) and the three output waveforms (‘vout_nmos’, ‘vout_pmos’, and ‘vout_txgate’). These are shown in the figure below. Visually, they look similar, but there’s more to them than meets the eye!



We will now plot the frequency spectrum (FFT) to visualize the linearity. For this, first select the signal for which you want to compute the FFT in the waveform window and then go to Measurements > Spectrum as shown in the figure below.





- Set the FFT Input method to “Calculate Stop Time”

- Earlier, we set the strobeperiod to 100n (this corresponds to a sample frequency of 10M). So, we set the Sample Freq to 10M.

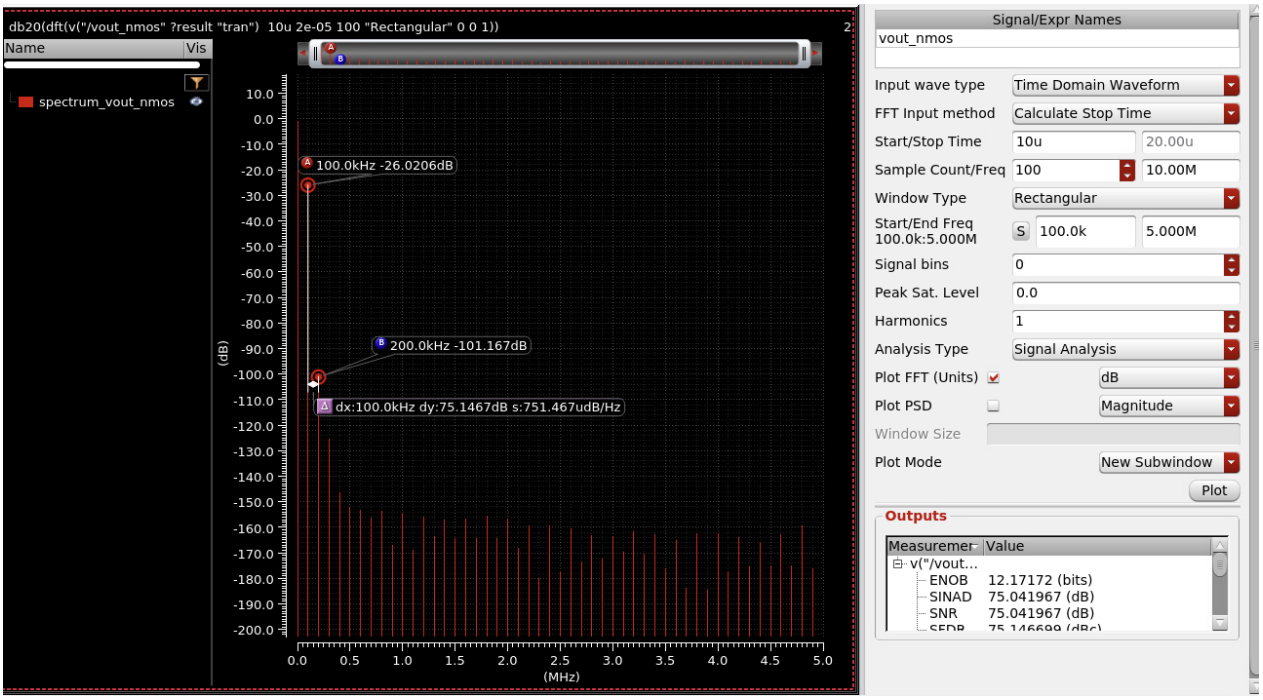
- We are interested in computing the FFT for 1 cycle of the input. This corresponds to 10u/100n = 100 samples. So, we set the Sample Count to 100.

- Finally, press S so that the Start/End Freq are automatically computed.

- Press Plot to plot the spectrum.

The frequency spectrum is as shown below. We note that there is a DC component (corresponding to our common-mode voltage), signal component at 100KHz, its harmonics at 200KHz, 300KHz, and so on. We note that these harmonics are because of the non-linearity of the switch.

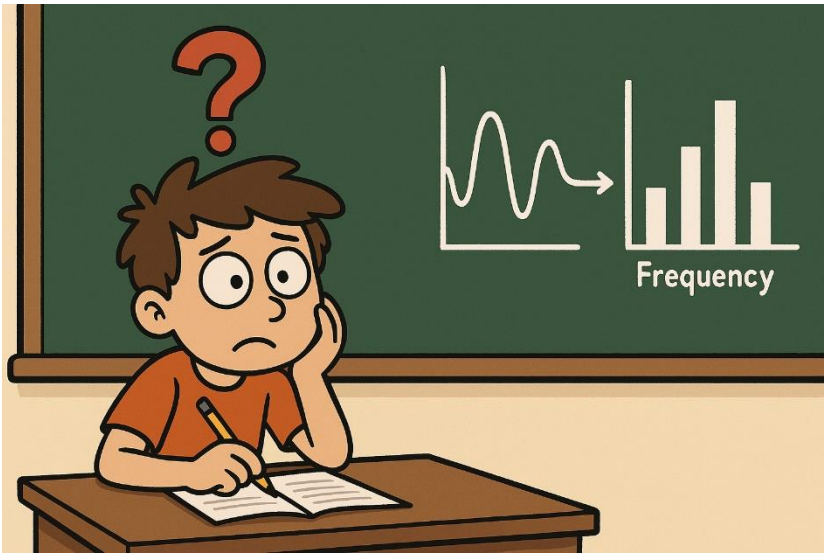
As a measure of the linearity, we will use the difference between the signal (at f_{in}) and the harmonics at $2f_{in}$ and $3f_{in}$. This is tabulated in the table below for the three switches.



Switch Configuration	Difference between tone at $2f_{in}$ and f_{in} (often referred to as Harmonic Distortion 2 or HD_2)	Difference between tone at $3f_{in}$ and f_{in} (often referred to as Harmonic Distortion 3 or HD_3)
NMOS	-75.1 dB	-99.3 dB
PMOS	-66.3 dB	-89.4 dB
TX Gate	-86.4 dB	-112.6 dB

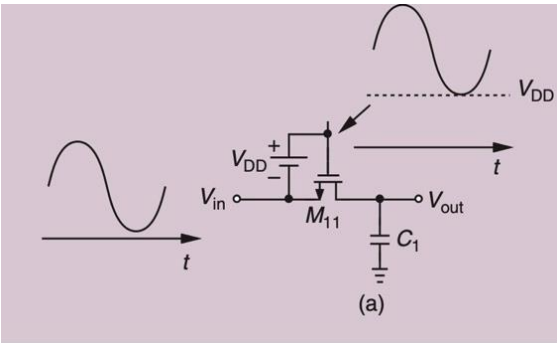
Repeat the experiment for vcm = 0.3V and 1.4V and construct a similar table. What are your observations?

Can you repeat the experiment with sampling? You can check with the TAs if you face any issue.



Take-home exercise (optional, but you better do it! ☺)

To reduce the variations of the on resistance of sampling switch between different samples, the bootstrapped switch can be used. The bootstrapped switch keeps the Vgs approximately constant



thereby minimizing the dependence of the switch resistance on the input signal. When the gate voltage tracks the input voltage (as shown in the Figure above), the dependence of the Vgs on input is reduced. For the complete version of the bootstrapped switch, we encourage you to read Prof. Razavi's paper in the SSCL magazine [1]. You are also encouraged to simulate the bootstrapped switch and check its linearity.

Reference

[1] https://www.seas.ucla.edu/brweb/papers/Journals/BRSummer15Switch.pdf