وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Ain Shams University – Faculty of Engineering – EECE Dept. – Integrated Circuits Lab.
Dr. Hesham Omran

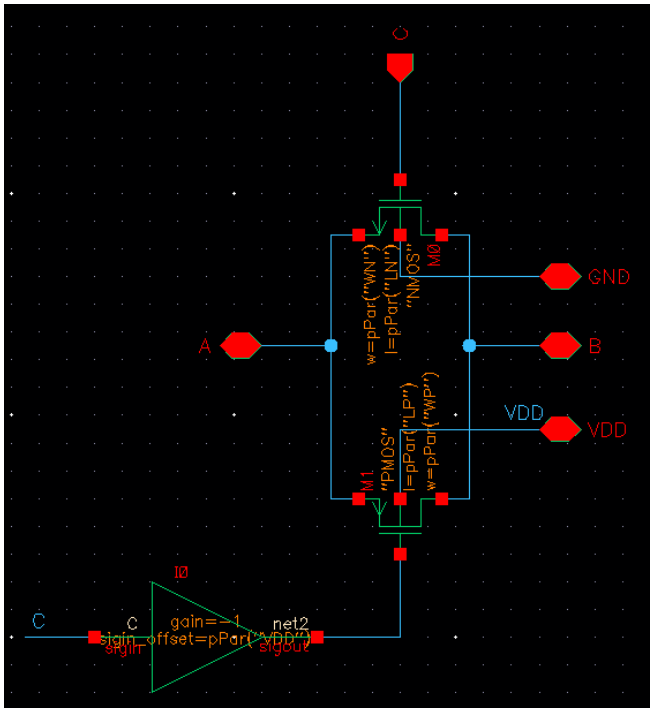
Analog Integrated System Design – Cadence Tools
Lab 04
Sample & Hold Circuits

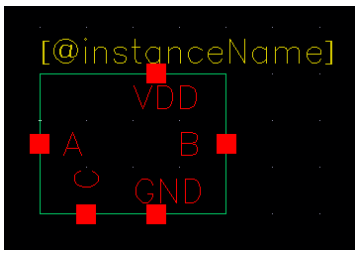
Lab Objective

1) To be familiar with the artifacts of S&H circuits.
2) To appreciate the importance of bottom-plate sampling.
3) To appreciate the importance of fully differential operation.

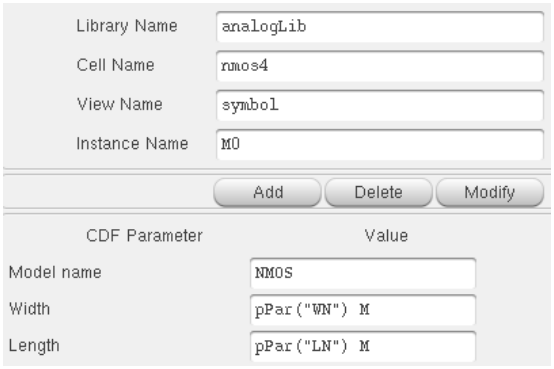
PART 1: S&H Artifacts

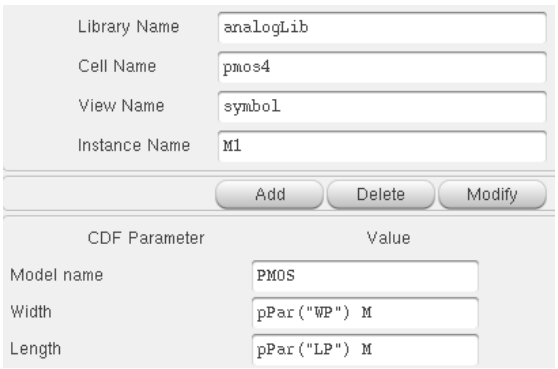
1) Create a new cell “tg” for a CMOS transmission gate (TG). Create schematic and symbol as shown below.



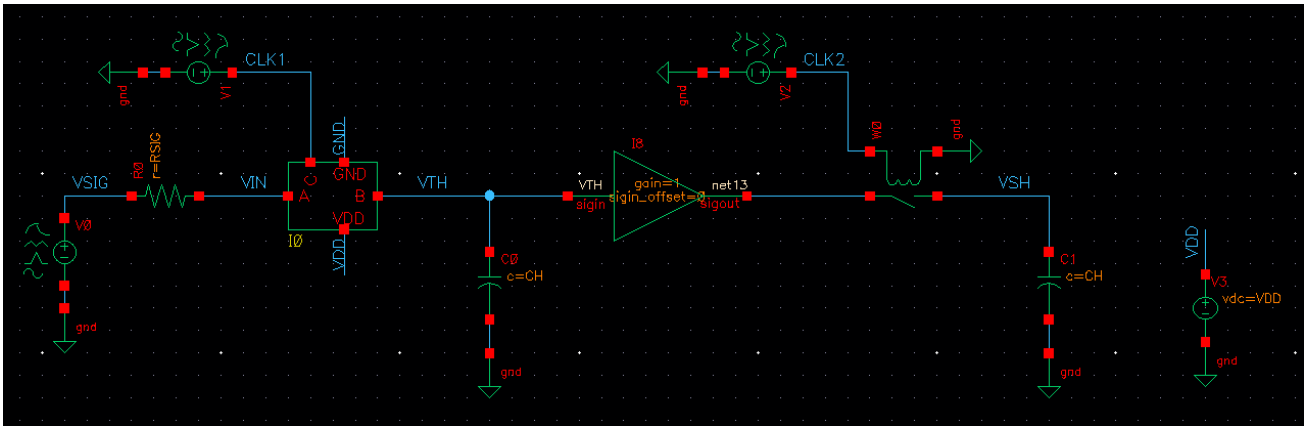


2) For the NMOS and PMOS use nmos4 and pmos4 from analogLib. Edit the “Model name” field in the properties to be NMOS for nmos4 and PMOS for pmos4. We will add the model file that contains the “NMOS” and “PMOS” models to adexl before we run the simulation as shown later.





3) For the ideal inverter use ahdlLib -> amp. Set gain = -1 and offset = pPar("VDD").
4) Copy Lab 02 Part 1 cell to a new cell and name it "lab_03_sah_tb". Modify the circuit as shown below. The schematic is similar to Lab 02 Part 1, but the first ideal switch is replaced by a transmission gate (TG). For the TG, set LN = LP = technology minimum and WN = WP = 10um.



5) For the ideal switch use analogLib -> switch
   - Open voltage: Relay resistance is 'ropen' at this voltage
   - Closed voltage: Relay resistance is 'rclosed' at this voltage

Open resistance	1T
Closed resistance	1
Open voltage	0.4*VDD
Closed voltage	0.6*VDD

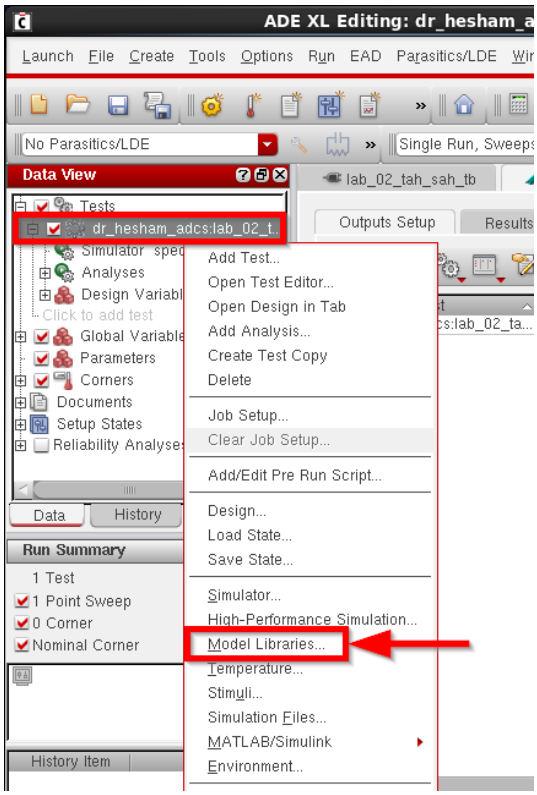
6) Set the input signal source as given below.

Type	Sin
Frequency	FIN
Amplitude	VPK
Offset (DC level)	VDC

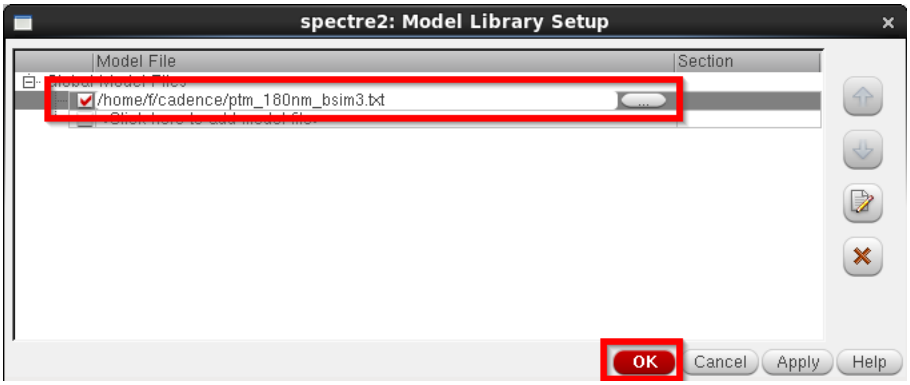
7) Set the clock signals as given below.

	CLK1	CLK2
Type	Pulse	Pulse
Zero value	0	0
One value	VDD	VDD
Period	TS	TS
Pulse width	TON	TON
Delay	0	0.5*TS
Rise/fall time	TRF	TRF

8) For the ideal buffer use ahdlLib -> amp. Set the gain to one and offset to zero.
9) Create adexl view. Create a new test. Right click on the test name and choose “Model Libraries”.



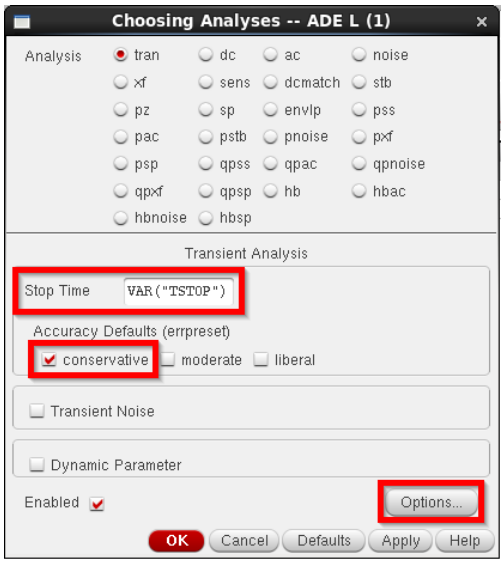
10) Browse to the PTM 180nm model file and add it to your model libraries as shown below.



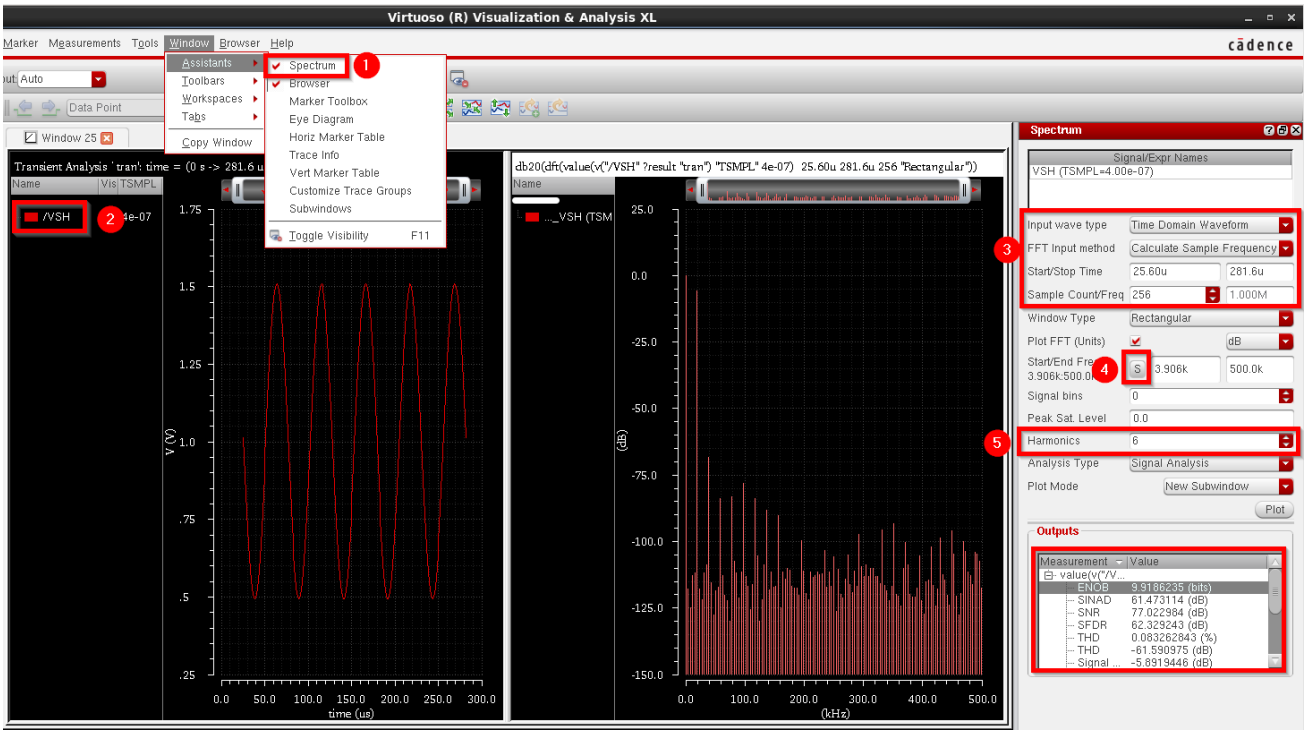
11) Import variables from schematic. Set the global variables as below. NCYC is the number of input signal cycles. NFFT is the number of FFT points. TS is the sampling period. TON is the T&H ON time (transparent window). TSTOP is extended by a half period (TDROP). This half period is dropped before doing the FFT to avoid simulator artifacts when simulation starts (and to avoid start-up artifacts in a real circuit). Note that NCYC, NFFT, and FIN are selected to satisfy the coherent sampling condition.

CH	1p
CP	0.1*CH
RSIG	1k
TS	1u
TON	0.4*TS
TRF	1n
NFFT	2**8
NCYC	5
FIN	(NCYC/NFFT)/TS
VDD	2
VDC	VDD/2
VPK	VDD/4
TDROP	0.5/FIN
TSTOP	NCYC/FIN + TDROP

12) Set transient simulation as below:



13) Run transient analysis. Plot VSIG and VTH overlaid. Zoom in to observe the S&H artifacts.
14) Use the Spectrum Assistant to plot FFT for VSH. Compare the results below with Lab 02 Part 1 results in a table. Comment on the differences.
    - ENOB
    - SINAD
    - SNR
    - SFDR
    - THD (in dB)
    - Signal power
    - DC power

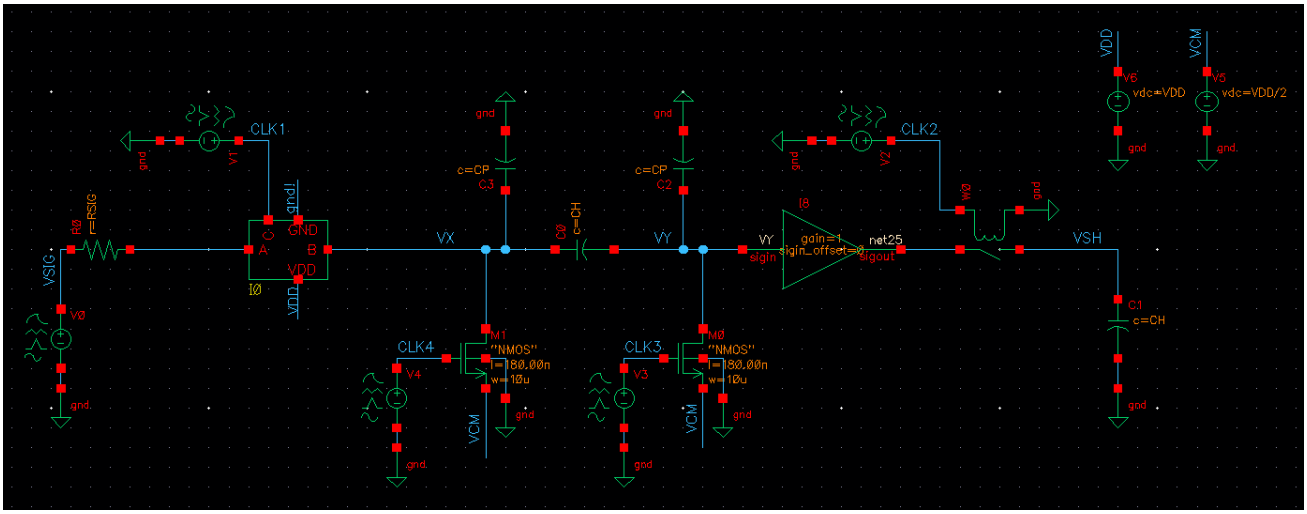


15) You can create the following expressions in the calculator and export them to adexl (or send them to adexl from the Spectrum Assistant) to quickly evaluate performance parameters.

Name	Type	Expression/Signal/File
spectrum_dcpower	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "dcpower")
spectrum_enob	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "enob")
spectrum_sfdr	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "sfdr")
spectrum_sigpower	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "sigpower")
spectrum_sinad	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "sinad")
spectrum_snb	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "snb")
spectrum_snr	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "snr")
spectrum_thddb	expr	spectrumMeasurement(getData("/VSH" ?result "tran") t 2.56e-05 0.0002816 256 3906 500000 0 "Rectangular" 0 0 7 "thddb")

PART 2: Bottom Plate Sampling

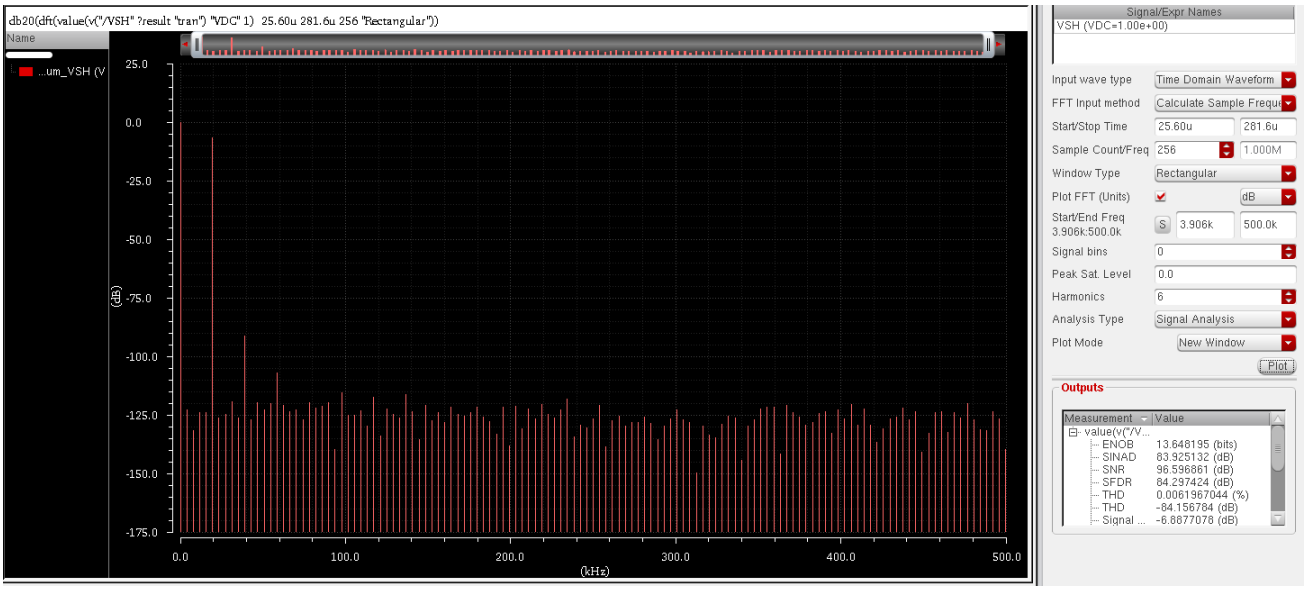
1) Copy the testbench to a new cell “lab_03_sah_bot_tb”. Construct the circuit shown below. Bottom plate sampling is used. The capacitors “CP” model the parasitic capacitors at VX and VY.



1) Set the clock signals as below.

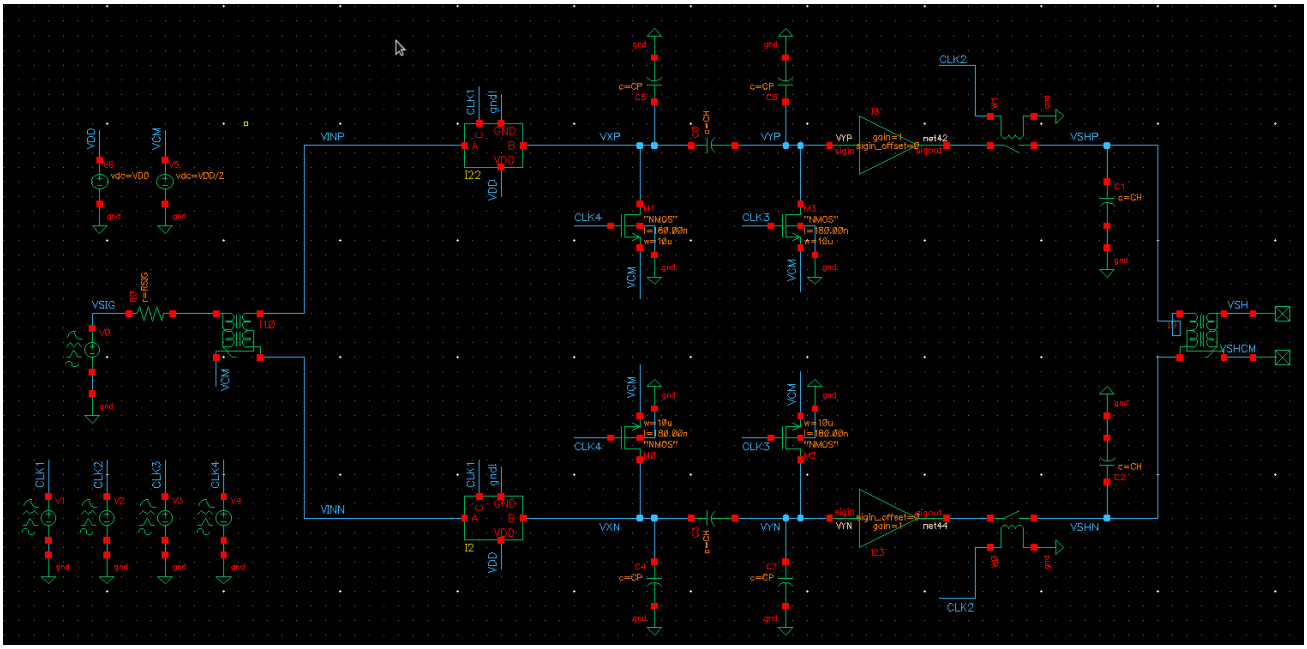
	CLK1	CLK2	CLK3	CLK4
Type	Pulse	Pulse	Pulse	Pulse
Initial value	0	0	0	VDD
Pulse value	VDD	VDD	VDD	0
Period	TS	TS	TS	TS
Pulse width	TON	TON	0.9*TON	1.1*TON
Delay	0	0.5*TS	0	0
Rise/fall time	TRF	TRF	TRF	TRF

2) Create adexl view. Add the PTM 180nm model file to your model libraries.
3) Import variables from schematic. Set the global variables and the transient simulation as in Part 1.
4) Run transient analysis. Observe the timing relations between different signals. Note that charge injection still exists, but it should be less independent on input signal (i.e., less non-linear) due to bottom-plate sampling.
5) Use the Spectrum Assistant to plot FFT. Compare the results below with Part 1 results in a table. Comment on the differences.
   - ENOB (Hint: The ENOB should improve significantly)
   - SINAD
   - SNR
   - SFDR
   - THD (in dB)
   - Signal power (Hint: The bottom plate parasitic capacitor attenuates the input signal)
   - DC power


db20(dft(value(v("/VSH" ?result "tran") "VDC" 1) 25.60u 281.6u 256 "Rectangular"))
Name
...um_VSH (V
25.0
0.0
-25.0
-50.0
-75.0
-100.0
-125.0
-150.0
-175.0
(dB)
0.0
100.0
200.0
300.0
400.0
500.0
(kHz)
Signal/Expr Names
VSH (VDC=1.00e+00)
Input wave type
Time Domain Waveform
FFT Input method
Calculate Sample Frequ
Start/Stop Time
25.60u
281.6u
Sample Count/Freq
256
1.000M
Window Type
Rectangular
Plot FFT (Units)
dB
Start/End Freq
3.906k:500.0k
S
3.906k
500.0k
Signal bins
0
Peak Sat. Level
0.0
Harmonics
6
Analysis Type
Signal Analysis
Plot Mode
New Window
Plot
Outputs
Measurement
Value
value(v("/V...
ENOB 13.648195 (bits)
SINAD 83.925132 (dB)
SNR 96.596861 (dB)
SFDR 84.297424 (dB)
THD 0.0061967044 (%)
THD -84.156784 (dB)
Signal ... -6.8877078 (dB)

PART 3: Fully Differential Operation

1) Copy the testbench to a new cell “lab_03_sah_bot_diff_tb”. Modify the schematic to be fully differential as shown below[1]. Use ideal_balun from analogLib.

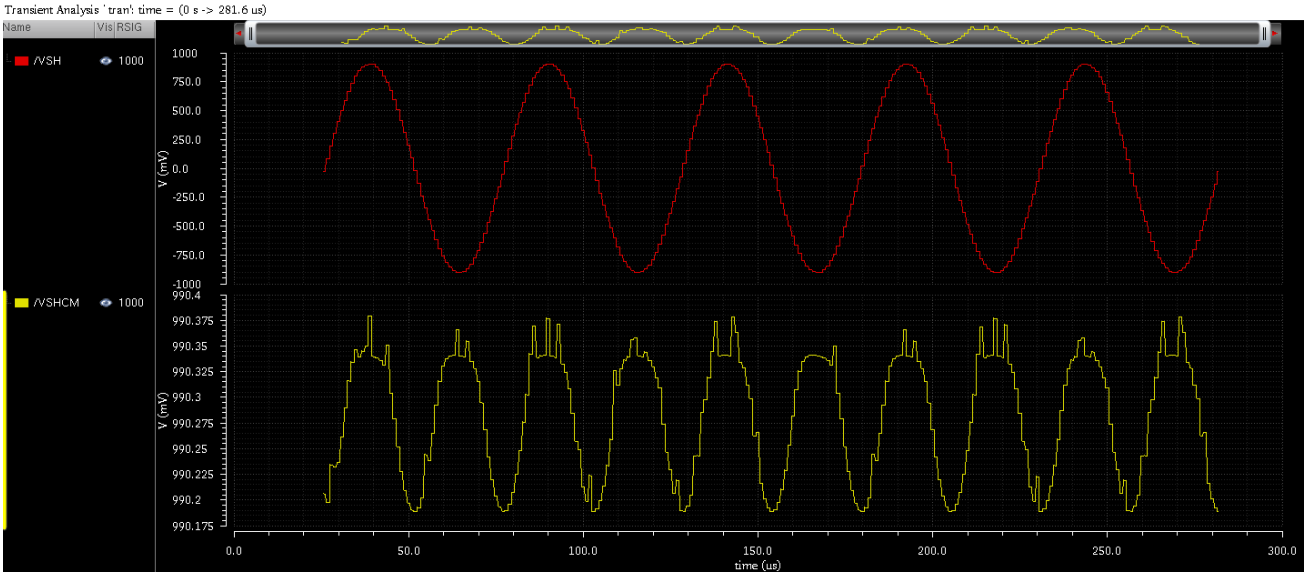


2) Modify the signal source to be as below. The differential operation doubles the signal swing (one extra bit).

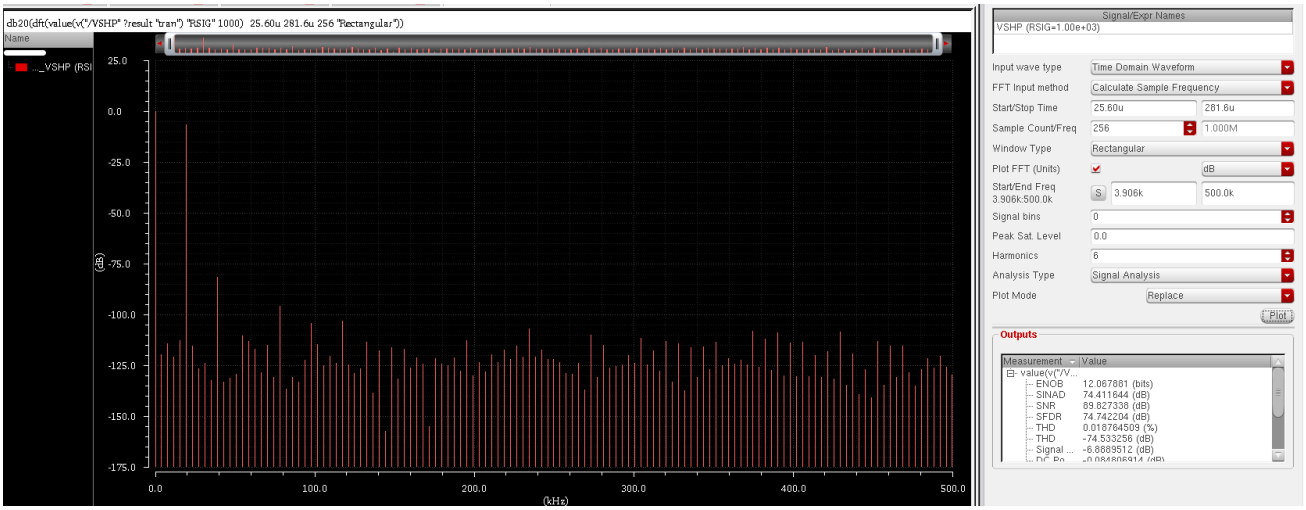
Type	Sin
Frequency	FIN
Amplitude	2*VPK
Offset (DC level)	**0**

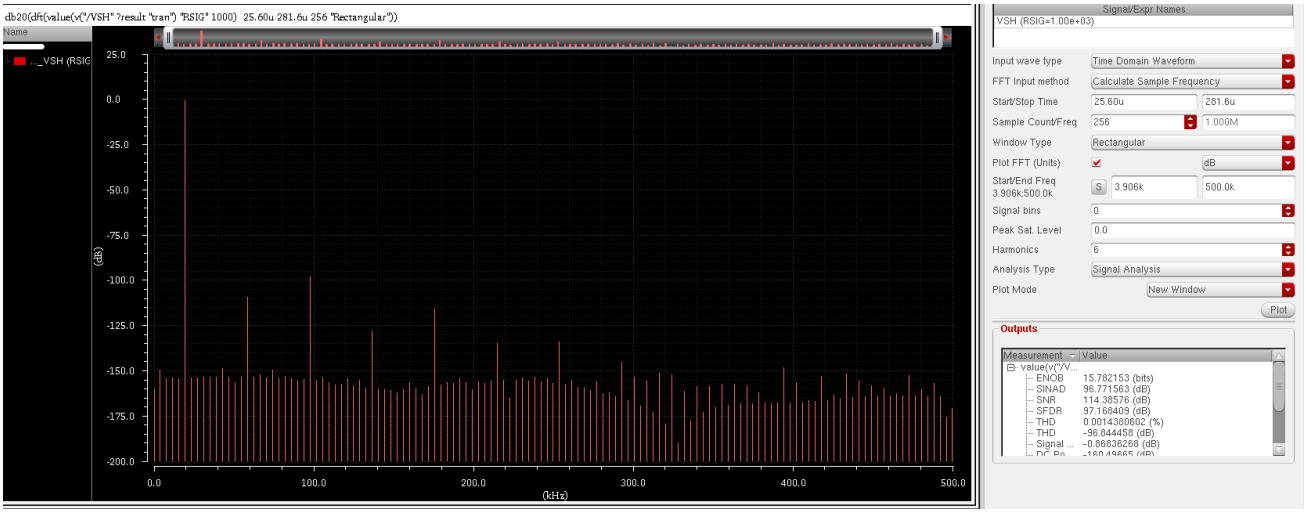
3) Plot the differential output and the common mode output vs time. Comment on the peak-to-peak differential output. Comment on the common output waveform.

[1] Increasing the parasitic cap Cp from 0.1pF to 1pF improves the ENOB by 2-bit. Cp mitigates charge injection errors but it attenuates the signal.



4) Use the Spectrum Assistant to plot FFT of the positive half output VSHP and the differential output VSH. **Compare the 2nd harmonic power in VSHP and VSH spectrum.** Compare the results below for VSHP and VSH in a table. Comment on the differences.
   - ENOB
   - SINAD
   - SNR
   - SFDR
   - THD (in dB)
   - Signal power
   - DC power




db20(dft(value(v("/VSH" ?result "tran") "RSIG" 1000) 25.60u 281.6u 256 "Rectangular"))
Name
..._VSH (RSIG
25.0
0.0
-25.0
-50.0
-75.0
-100.0
-125.0
-150.0
-175.0
-200.0
(dB)
0.0
100.0
200.0
300.0
400.0
500.0
(kHz)
Signal/Expr Names
VSH (RSIG=1.00e+03)
Input wave type
Time Domain Waveform
FFT Input method
Calculate Sample Frequency
Start/Stop Time
25.60u
281.6u
Sample Count/Freq
256
1.000M
Window Type
Rectangular
Plot FFT (Units)
dB
Start/End Freq
3.906k:500.0k
S
3.906k
500.0k
Signal bins
0
Peak Sat. Level
0.0
Harmonics
6
Analysis Type
Signal Analysis
Plot Mode
New Window
Plot
Outputs
Measurement
Value
value(v("/V...
ENOB
15.782153 (bits)
SINAD
96.771563 (dB)
SNR
114.38576 (dB)
SFDR
97.168409 (dB)
THD
0.0014380602 (%)
THD
-96.844458 (dB)
Signal ...
-0.66836268 (dB)

PART 4 (Optional): Bootstrapping

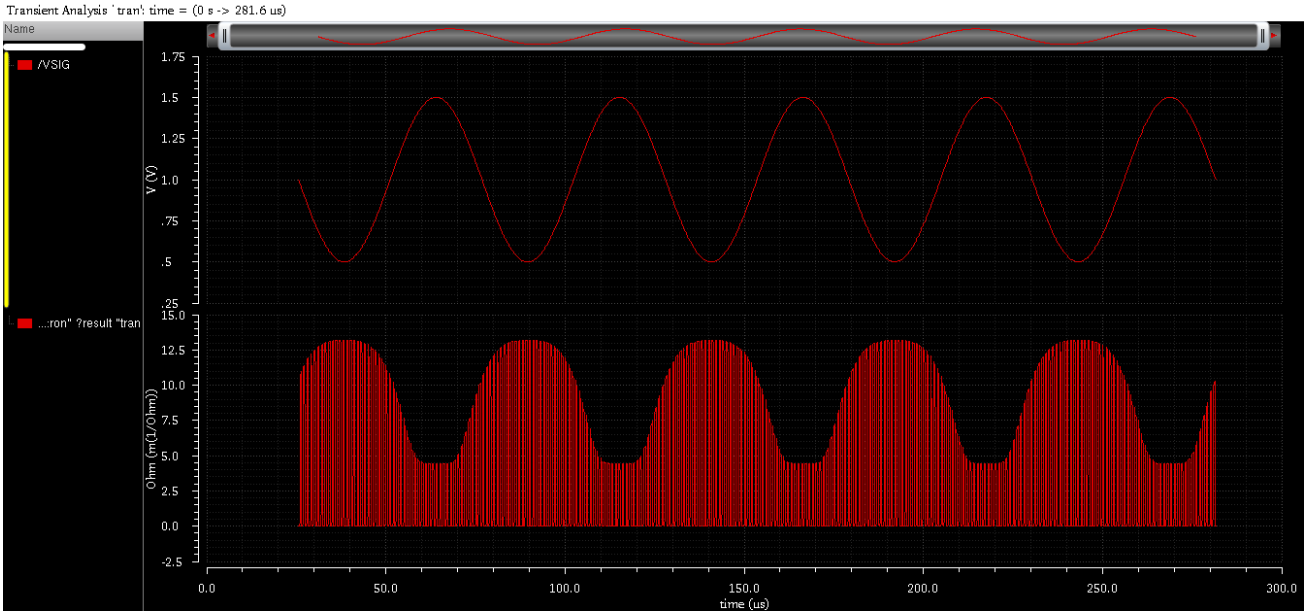
1) Go back to the testbench of PART 1. We want to plot the variation of the sampling switch on resistance vs the input signal. We want to save the parameter “ron” of the CMOS TG. Create a new text file “save_tran.txt” and write the following (change the name of the cell and the transistor to match your schematic):

```
simulator lang = spectre
save I0.M0:ron sigtype=dev
save I0.M1:ron sigtype=dev
```

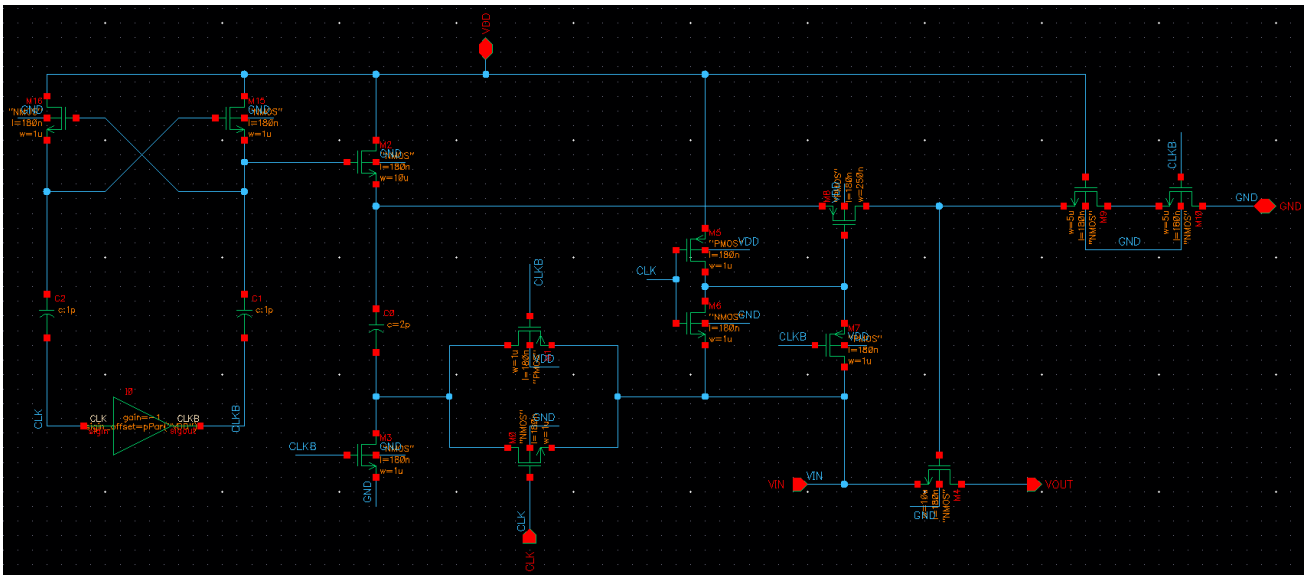
2) In adexl right click on the test name, choose Model Libraries, and add the text file you just created.
3) Run the transient simulation. Use the calculator to plot the CMOS TG conductance using the expression below:

```
1/getData("I0.M0:ron" ?result "tran") + 1/getData("I0.M1:ron" ?result "tran")
```

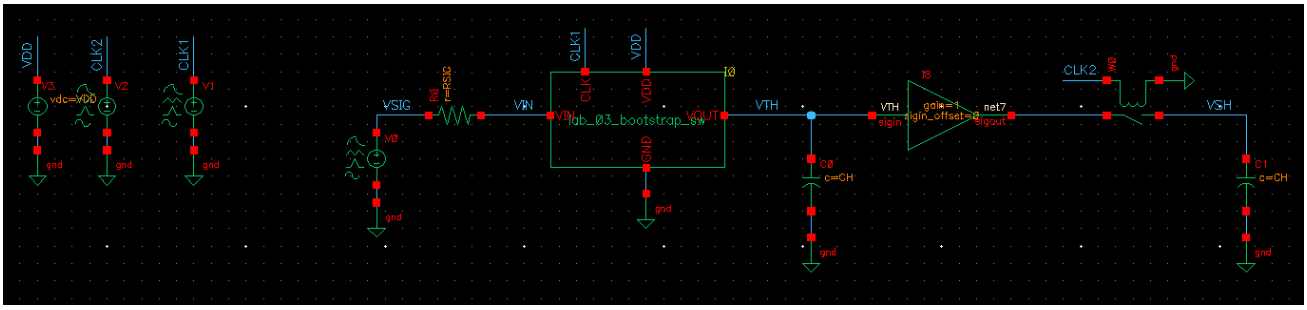
4) Notice how the conductance varies with the input signal (non-linear behavior). Calculate the percent of variation = peak-to-peak variation / average.



5) Record the ENOB.
6) Create a new schematic for a bootstrapped switch as shown below.



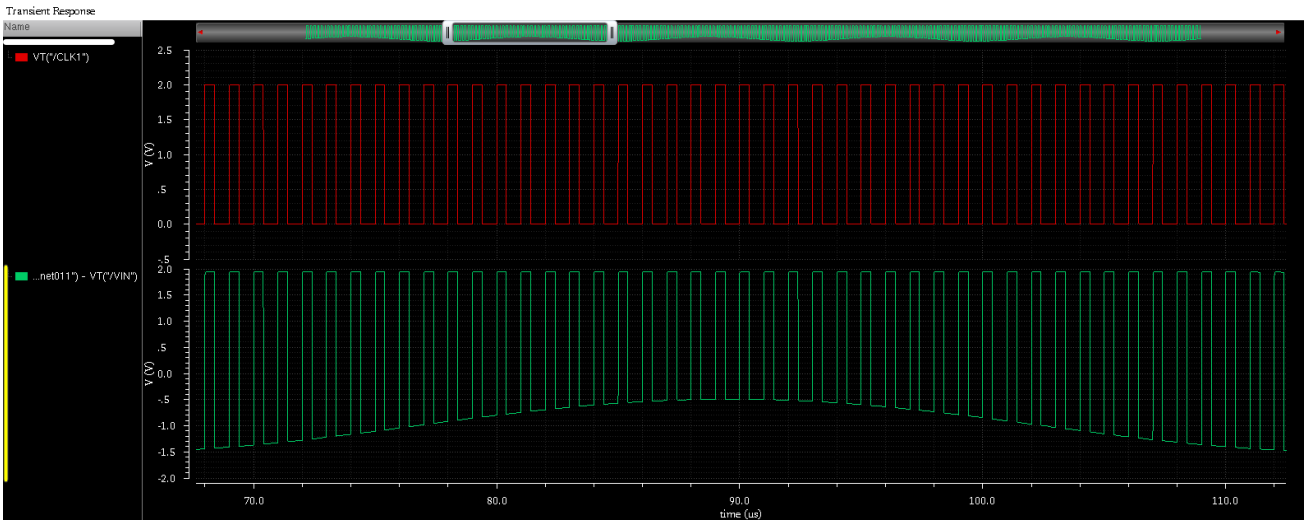
7) Create a new testbench for the S/H using the bootstrapped switch “lab_03_sah_bootstrap_tb”.



8) Add a text file to Model Libraries to save the sampling switch “ron” (change the name of the cell and the transistor to match your schematic):

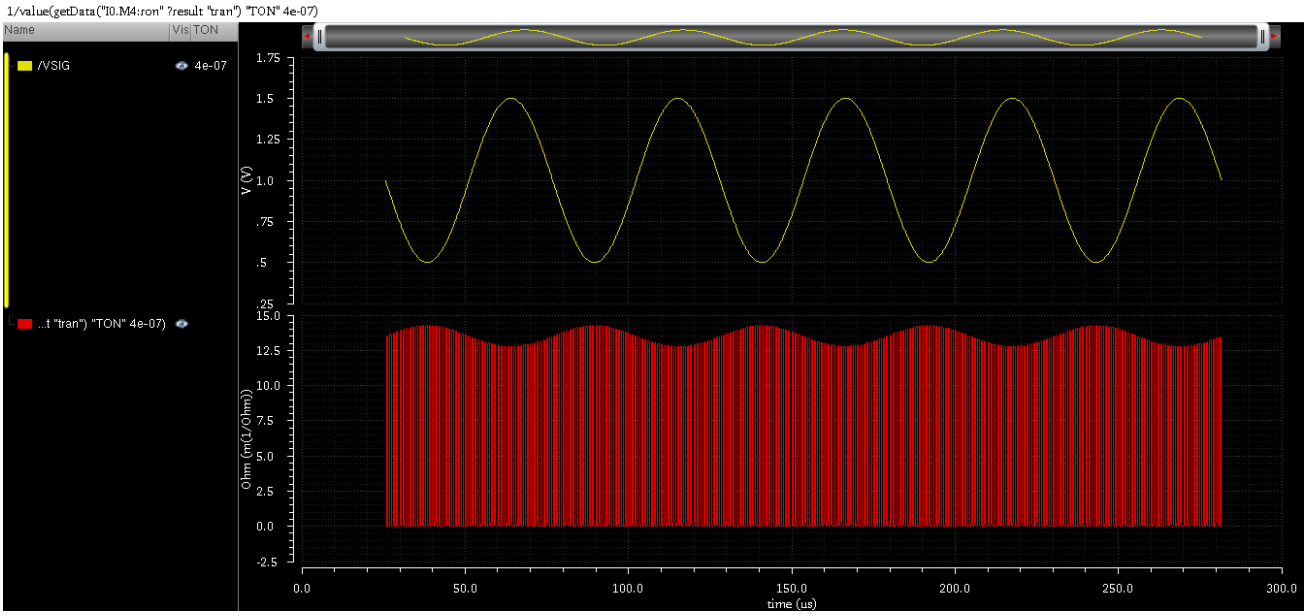
```
simulator lang = spectre
save I0.M4:ron sigtype=dev
```

9) Run the transient simulation. Investigate the transient waveforms at different nets of the bootstrapped switch and understand how it works. Plot and examine the VGS of the sampling switch. Does VGS depend on the input signal when the switch is in the ON state?



10) Use the calculator to plot the bootstrapped switch conductance. Calculate the percent of variation = peak-to-peak variation / average. Compare it to the previous case.

11) Ideally the conductance should not vary. Why there is still some variation?



12) Compare the ENOB to the previous case? Is it improved? Why the ENOB is not affected by the switch non-linearity? (Hint: compare the switch on resistance to the signal source resistance and compare the signal frequency with S/H tracking bandwidth).