وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا

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

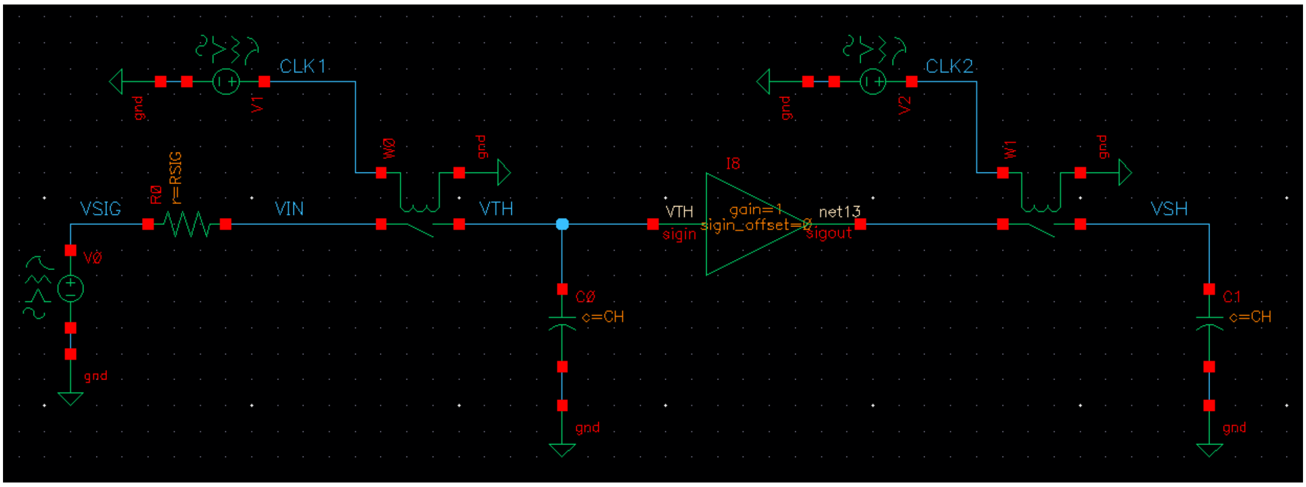
Analog Integrated System Design – Cadence Tools
Lab 02
Sampling and Quantization

Lab Objective

1) To be familiar with the operation of simple T&H and S&H circuits.
2) To be able to do spectral analysis and calculate various performance metrics using FFT.
3) To understand and apply the coherent sampling condition.
4) To calculate the effect of FFT number of points on the noise floor.
5) To be familiar with behavioral modeling and Verilog-A.
6) To be familiar with the effect of quantization on SNR.

PART 1: Ideal Track & Hold and Sample & Hold

1) Create a new cell “lab_02_tah_sah_tb”. Construct the circuit shown below. The schematic consists of two T&H circuits separated by an ideal buffer. Note that if we don’t use a buffer, charge sharing will occur between the two hold capacitors.



2) 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

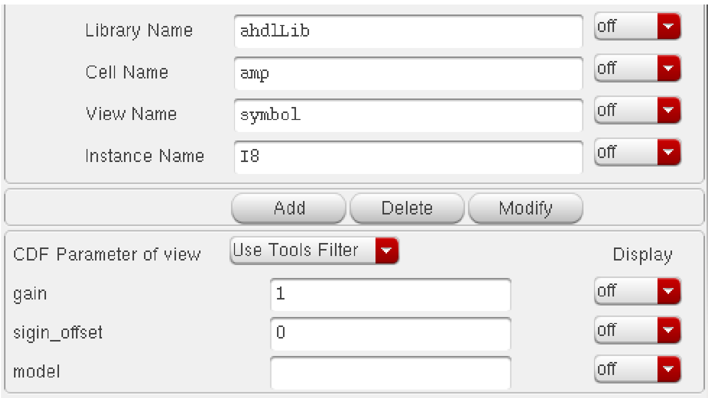
3) Set the input signal source as given below.

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

4) 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

5) For the ideal buffer use ahdlLib -> amp.

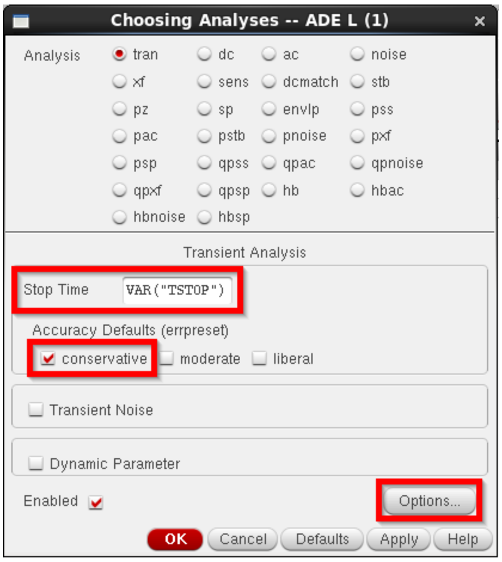


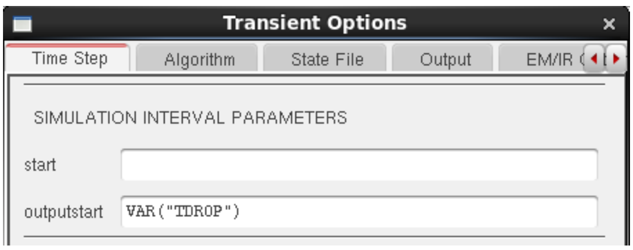
6) Create adexl view. Import variables from schematic. Set the global variables as below. NCYC is the number of input signal cycles. NFFT is the number of FFT points. TS is the sampling period. TON is the T&H ON time (transparent window). TSTOP is extended by a half period (TDROP). This half period is 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

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

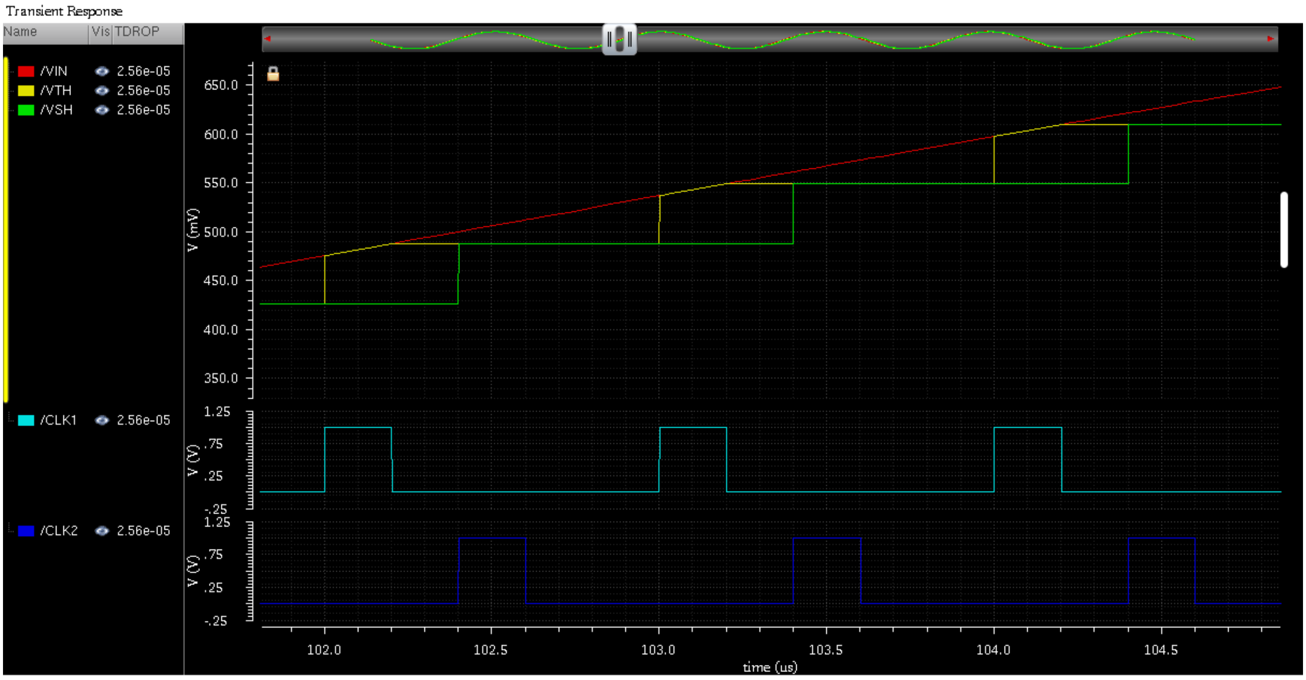
7) Set transient simulation as below.



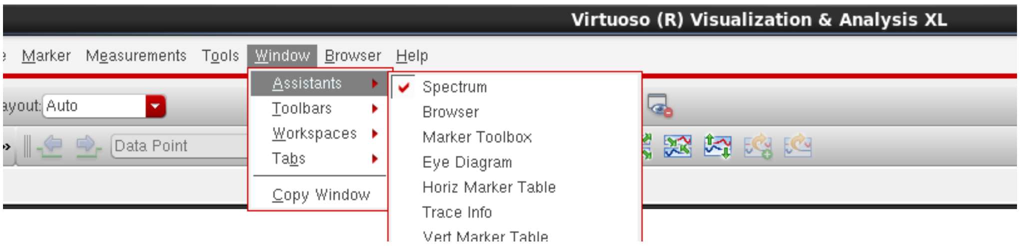


8) Run transient analysis. Plot VSIG, VTH, and VSH overlaid. Zoom in to observe the difference between T&H (VTH) and S&H (VSH).

[1] Note that "**" is the exponent operator; thus, 2**8 = 256.

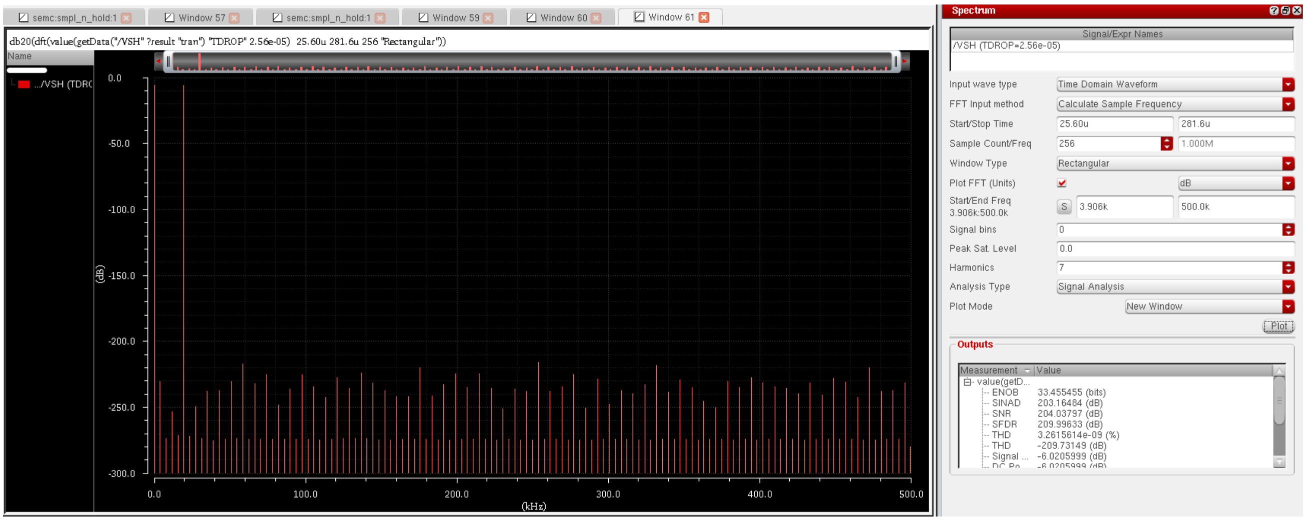


9) Use the Spectrum Assistant to plot FFT. Familiarize yourself with the different options in the Spectrum Assistant window.
   - What is the power of the peak signal (in dB)? Why?[2]
   - How many bins are occupied by the test signal?
   - What is the noise floor (in dBFS)?
   - What is the relation between the SNR, NFFT, Signal Power, and Noise Floor?
   - If the sampling is ideal, what is the source of error that causes the noise floor? [3]



[2] Note that the output of the DFT is the signal amplitude. If you plot the spectrum in dB it will plot 20*log(amplitude) not 20*log(rms).

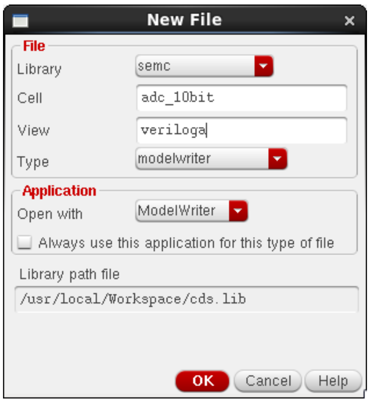
[3] If the ENOB is significantly less than the value reported in the snapshot, you may try setting a tight "maxstep" in the transient simulation settings.

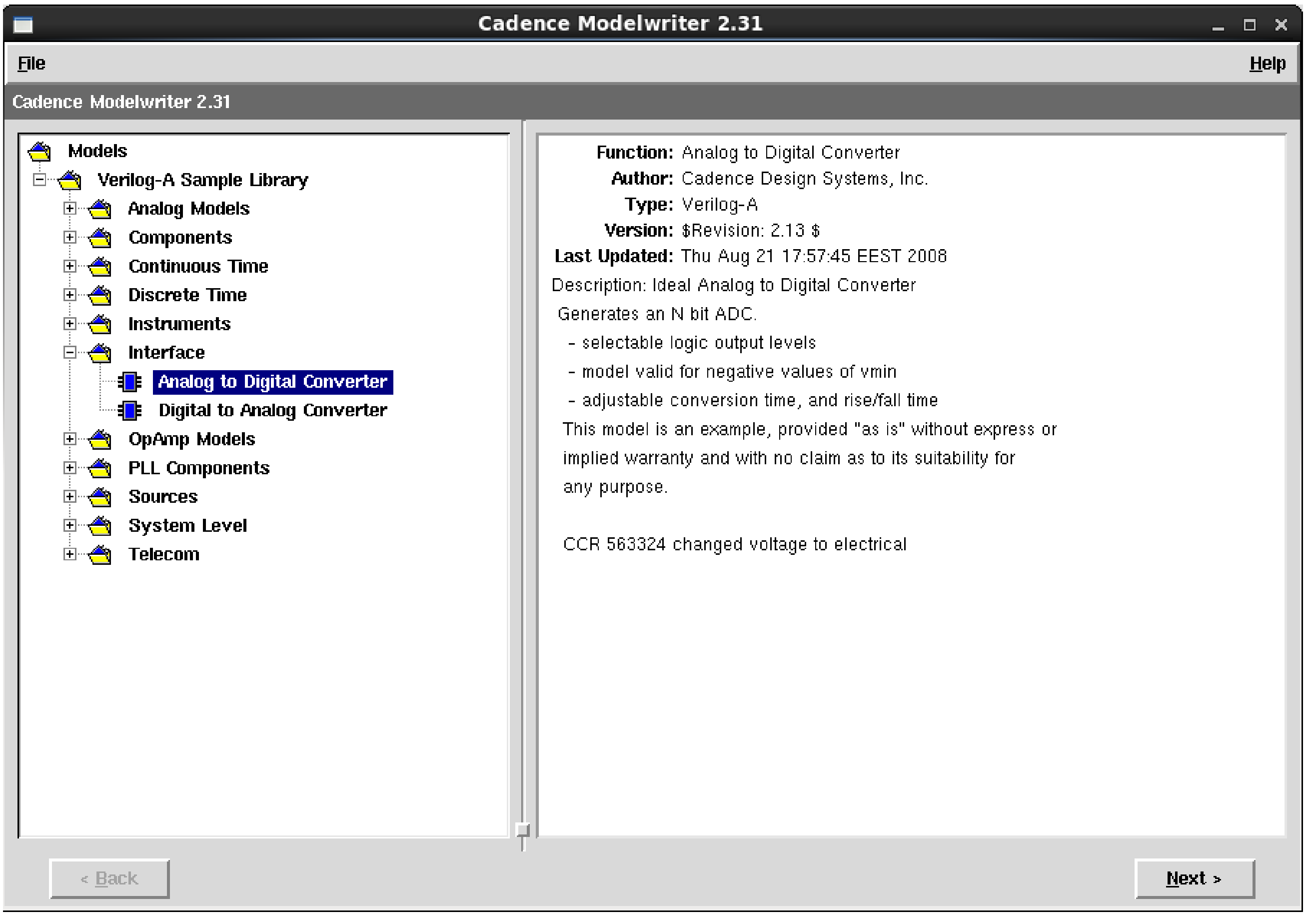


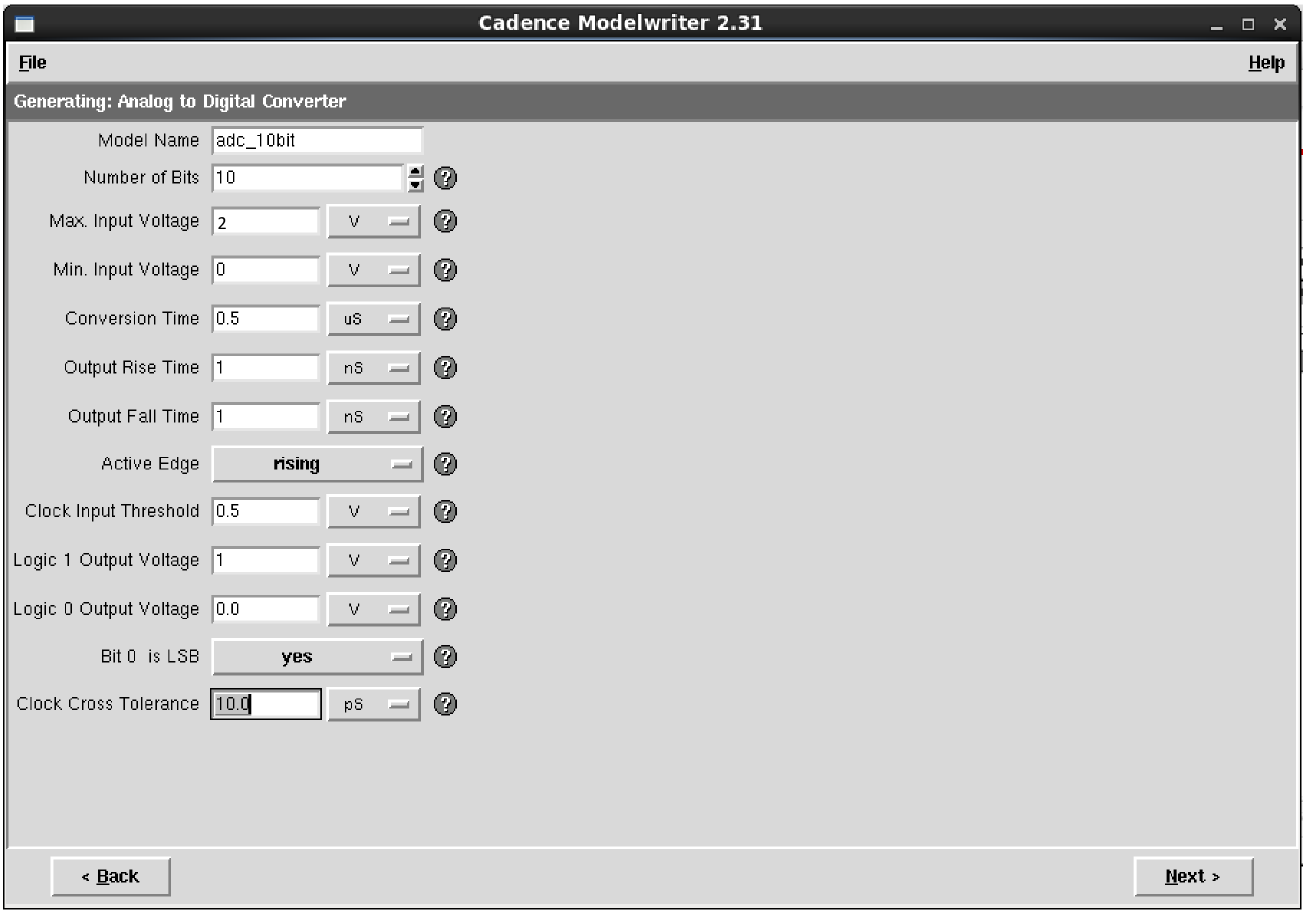
10) Change NCYC to 5.5 and re-simulate. Note that the start and stop time in the DFT will change from the previous case. Plot the new FFT. Observe the spectral leakage.

PART 2: Quantization

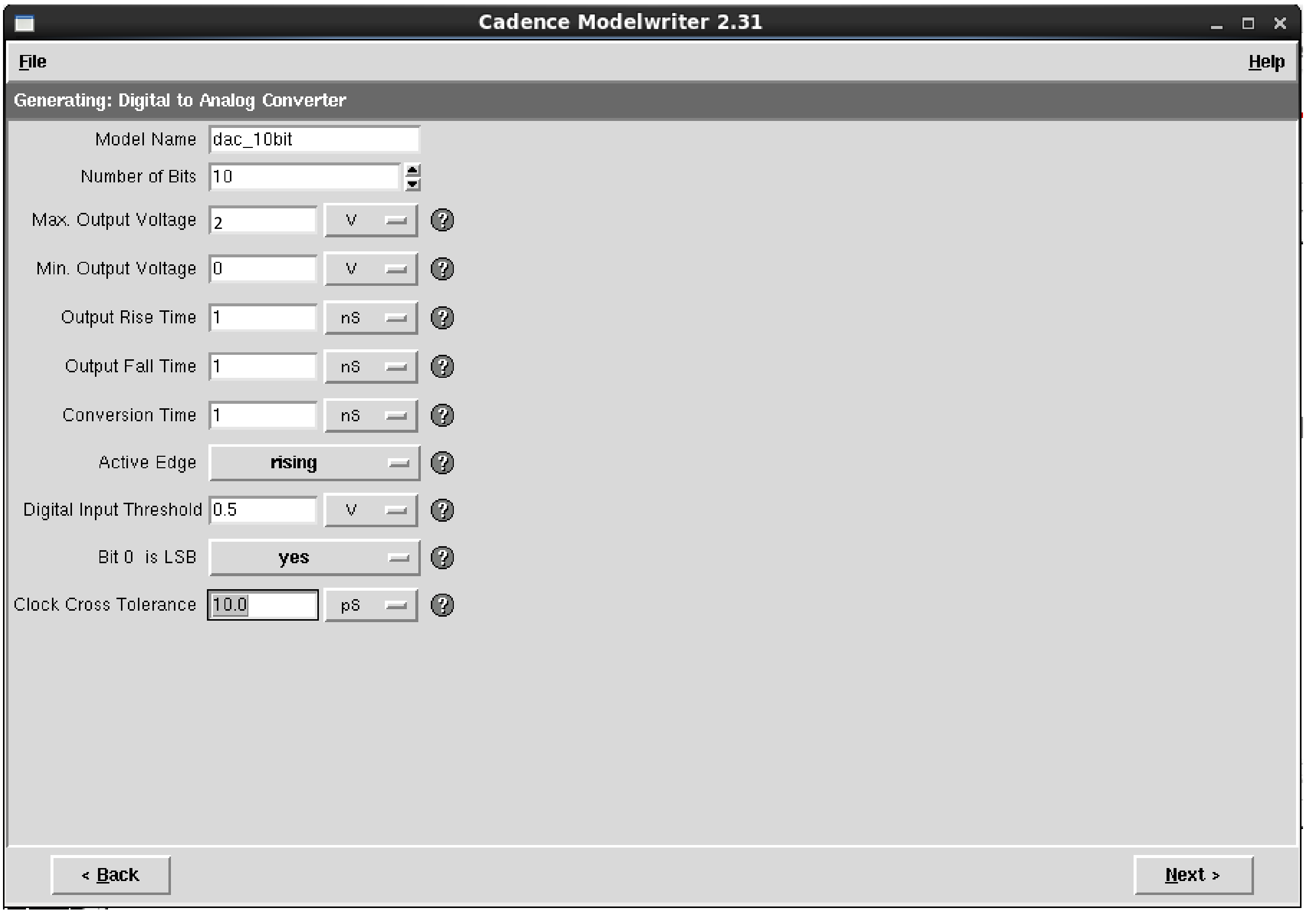
1) Use Modelwriter to create a veriloga model for a 10-bit ADC. Create a symbol for the generated view.



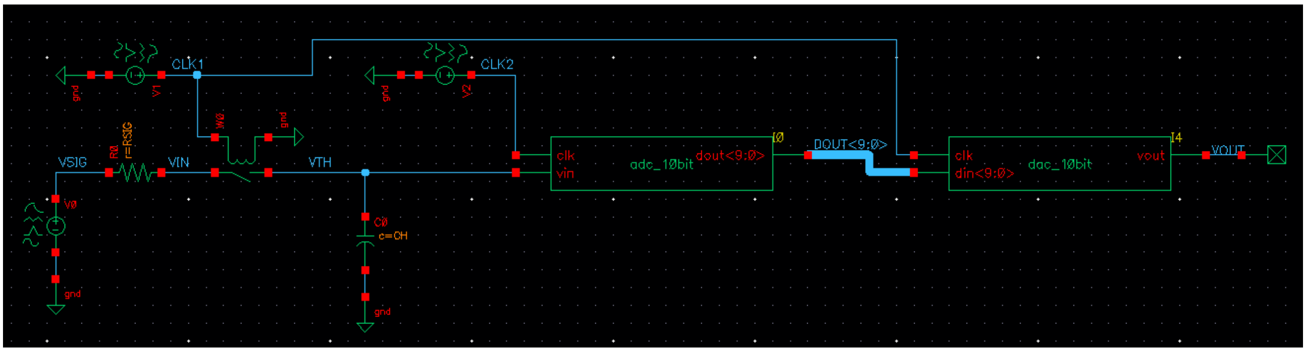

Cadence Modelwriter 2.31
File
Help
Cadence Modelwriter 2.31
Models
Verilog-A Sample Library
Analog Models
Components
Continuous Time
Discrete Time
Instruments
Interface
Analog to Digital Converter
Digital to Analog Converter
OpAmp Models
PLL Components
Sources
System Level
Telecom
Function: Analog to Digital Converter
Author: Cadence Design Systems, Inc.
Type: Verilog-A
Version: $Revision: 2.13 $
Last Updated: Thu Aug 21 17:57:45 EEST 2008
Description: Ideal Analog to Digital Converter
Generates an N bit ADC.
- selectable logic output levels
- model valid for negative values of vmin
- adjustable conversion time, and rise/fall time
This model is an example, provided "as is" without express or
implied warranty and with no claim as to its suitability for
any purpose.
CCR 563324 changed voltage to electrical
< Back
Next >


Cadence Modelwriter 2.31
File
Help
Generating: Analog to Digital Converter
Model Name adc_10bit
Number of Bits 10
Max. Input Voltage 2 V
Min. Input Voltage 0 V
Conversion Time 0.5 uS
Output Rise Time 1 nS
Output Fall Time 1 nS
Active Edge rising
Clock Input Threshold 0.5 V
Logic 1 Output Voltage 1 V
Logic 0 Output Voltage 0.0 V
Bit 0 is LSB yes
Clock Cross Tolerance 10.0 pS
< Back
Next >

2) Similarly, use Modelwriter to create a veriloga model for a 10-bit DAC. Create a symbol for the generated view.

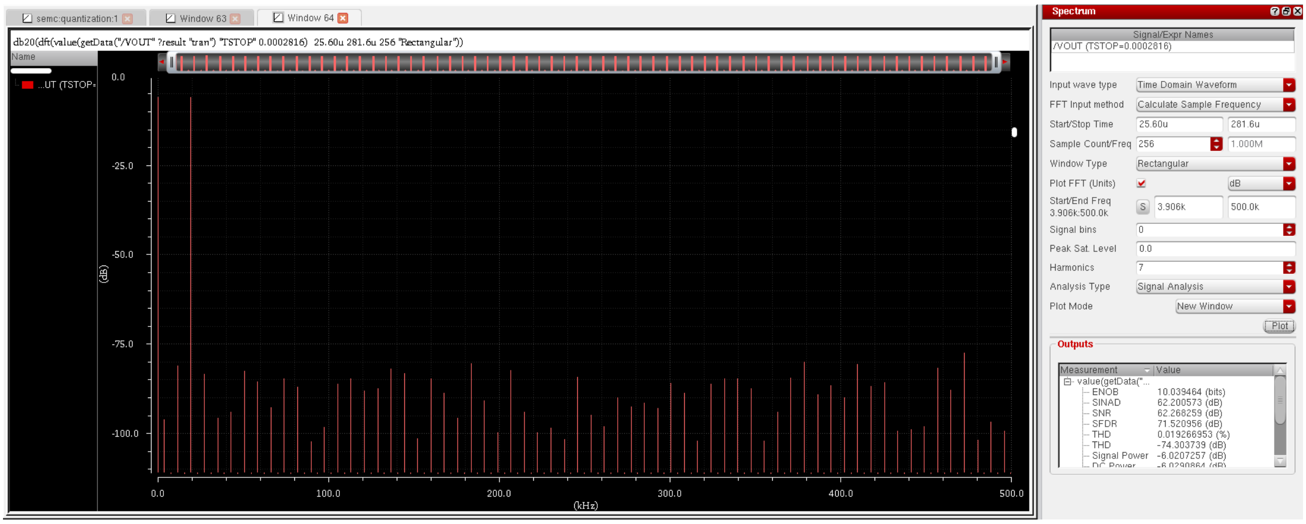


3) Create a new testbench “lab_02_quant_tb” as shown below.

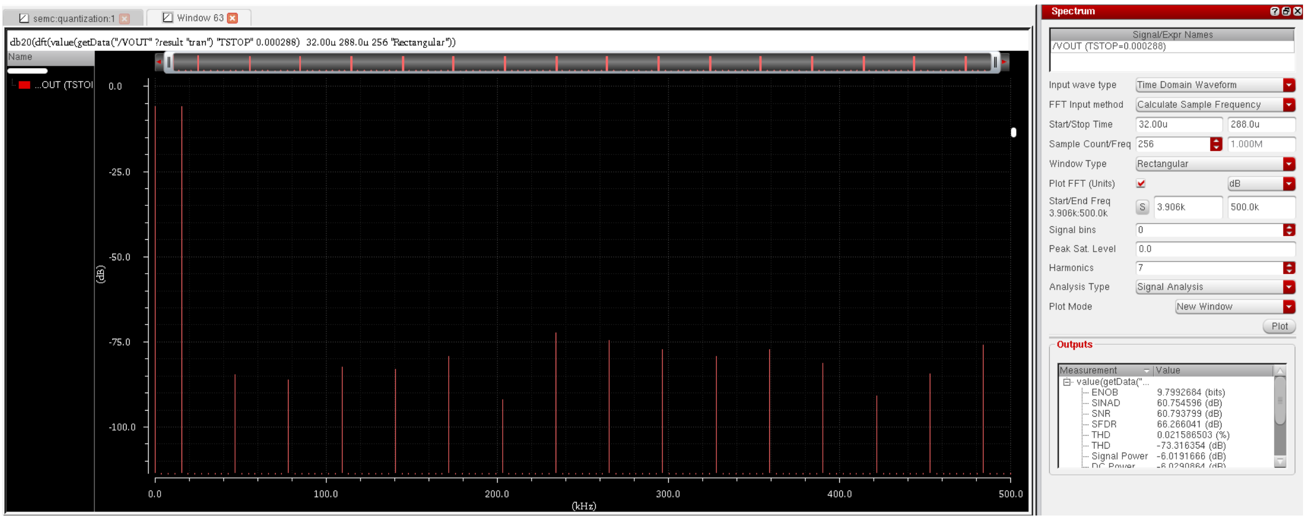


4) Create a new adexl view. Set global variables as in Part 1. Set transient analysis as in Part 1.
5) Plot the transient waveforms and study the timing relations between different signals.
6) Analyze the DAC output using the spectrum assistant. The result will be as shown below (zoom in y-axis from 0 to -100dB). Compare the SNR, ENOB, Signal Power, DC Power, and Noise Floor with the expected theoretical values.[4]

[4] You will need to increase the input signal amplitude, otherwise, the ENOB will be one bit less than the expected ideal value (why?).



7) Record the value of the SFDR.
8) Change NCYC to 4 and re-simulate (now NFFT/NCYC = integer). Plot the new FFT (zoom in y-axis from 0 to -100dB). Note that the start and stop time in the DFT will change from the previous case. Compare the new SFDR with the previous one. Comment.



Appendix

If you don't have Spectrum Assistant (available in IC 6.1.6) in your Cadence version, you may do post-processing in Matlab.

Perform DFT in cadence then export the DFT result to csv file:

```
dB20(dft(VT("/VOUT") VAR("TDROP") VAR("TSTOP") VAR("NFFT") "Rectangular" 1 "default" ))
```

Use the following code in Matlab (edit the paths):

```
% original code by B. Boser (Berkeley)
% Edited by H. Omran (ASU)
clear; clc;
% load the cvs file
% the file has the freq and DFT of the signal in dB
M = csvread('lab3_part2_6.csv',1,0);
f = M(:,1);
sdb_cadence = M(:,2);
NCYC = 5; % # of signal cycles
NFFT = 2^8; % # of FFT points
s = 10.^(sdb_cadence/20); % DFT of output signal in volt
Ts = 1; % normalized period
```

```
fx = NCYC/NFFT/Ts; % freq of the input signal
fs = 1/Ts;
N = NFFT;
bx = N * fx/fs + 1; % signal bin (the index for the signal in freq domain)
fh = (2:8) * fx; % harmonic freqs
% Get the alising of harmonic signals
while max(fh) > 0.5
fh = abs(fh - (fh>0.5));
end
bh = N * fh / fs + 1; % harmonic bins
Asignal = s(bx); % signal amplititude in freq domain
sn = s; % Noise
sn(bx) = 1e-100; % nulling signal bin
sn(1) = 1e-100; % nulling DC bin
SFDR = 20*log10(Asignal/max(sn));
Aharm = sqrt(sum(sn(bh).^2));
sn(bh) = 1e-100;
Anoise = sqrt(sum(sn.^2));
NoiseFloor = 10*log10(sum(sn.^2)/(NFFT/2-length(bh)-2));
SNR = 20*log10(Asignal/Anoise);
SNDR = 10*log10(Asignal.^2 /(Anoise.^2 + Aharm.^2));
THD = -20*log10(Asignal/Aharm);
ENOB = (SNDR - 1.76)/6.02;
figure;
clf;
sdb = 20*log10(s);
plot(f, sdb, 'b-');
hold on;
plot(f(bx), sdb(bx), 'ro');
plot(f(1), sdb(1), 'y+');
plot(f(bh), sdb(bh), 'gx');
for i=1:3
x = bh(i);
text(f(x), sdb(x), sprintf(' H_{%d} = %.1fdBFS', i, sdb(x)));
end
text(f(1), sdb(1)-10, sprintf(' DC = %.1fdBFS', sdb(1)));
text(f(bx), sdb(bx), sprintf(' A = %.1fdBFS', sdb(bx)));
xlabel('Frequency [ f / f_s ]');
ylabel('Amplitude [ dBFS ]');
title(sprintf('N = %d ENOB = %.1f-bit SNR = %.1fdB THD = %.1fdB SNDR = %.1fdB SFDR = %.1fdB NoiseFloor =
%.1fdB', N, ENOB, SNR, THD, SNDR, SFDR, NoiseFloor));
axis([ min(f) max(f) -150 10 ]);
hold off;
```