Assignment 1

1. In this assignment, we develop a behavioral model of an ideal sample and hold amplifier in VerilogA using the Cadence IC5.0 suite and plot the spectral response in MATLAB.

Sample and hold
Create a VerilogA S/H model based on the "sha_ideal" cell (Library ahdlLib, view veriloga). Set the clock transition voltage to be 1.5V. Create a symbol for your S/H cell. Instantiate the S/H model to a new cell "testbench" (schematic).
For VerilogA reference see $CAD_HOME/ic50/doc/veriaref/veriaref.pdf

Test input
Apply a sine input of frequency = 1 KHz, amplitude = 1V and a dc voltage = 1.5V. Use a clock of 10us period (100KHz) with 50% duty cycle, 10 ns rise/fall time and amplitude = 0-3V.

Simulation parameters
Simulation time = 1/ (Resolution required (Hz)). Here, we target a resolution of 100Hz. Therefore, simulation time = 10ms
Time step depends on the highest frequency. For a frequency of 100 KHz, time step = 10µs. Typically we capture 4 data points per cycle, i.e. every 2.5µs.
Thus we have a total number of 10ms/2.5µs = 4000 points.
For the FFT, we round off the number of points to the nearest power of 2, which, in this case is 2^{14} = 4096. For the 96 extra points, we simulate for an additional 96*2.5µs = 240µs giving a total simulation time = 10.24ms

Simulation
For an accurate simulation, the simulator time step is set to 1% of the required time step. Run a transient analysis using Spectre and under options set step = maxstep = 0.01*10µs = 100ns.
Plot the sampled output waveform.
Type the following OCEAN command in the CIW window to capture the data points in a file.
ocnPrint(VT("/output_node") ?output "./filename" ?numberNotation 'scientific ?from 2.5u ?to 10.24m ?step 2.5u)

MATLAB
Edit the output file to remove the headers. Import the output file into MATLAB and use the following code fragment to plot the spectrum.

```
n = 4096;
y = filename(:,2);
mid = ceil(n / 2) + 1;
wc = blackman(n);      %%wc are the window coefficients
win = y.* wc';         %%win is the windowed input
```

```
wvfft = fftshift(fft(win,n)/length(win));
winpower = 2*(abs(wvfft)).^2;
winpower_single = winpower(mid:n); %single sided signal power spectra

% single sided FFT
fpts = (2 * pi / n) * [0:(n-1)];           %%0 to 2pi
fpts(mid:n) = fpts(mid:n) - 2 * pi;        %%0 to pi, -pi to 0
fpts = fftshift(fpts);                     %%-pi to pi
freq=fpts(mid:n)/pi * fs/2;

% Plot power spectra
figure;
semilogx(freq, 10*log10(winpower_single));
xlabel('Frequency: (Hz)');
ylabel('Power: (dB)');
```

Verify the sinc attenuation = $20\log(\frac{\pi.1KHz)}{100KHz)}$ from the plot.

Try the above exercise for different input frequencies/clock period and other windows such as rectangular, hamming etc.

2.