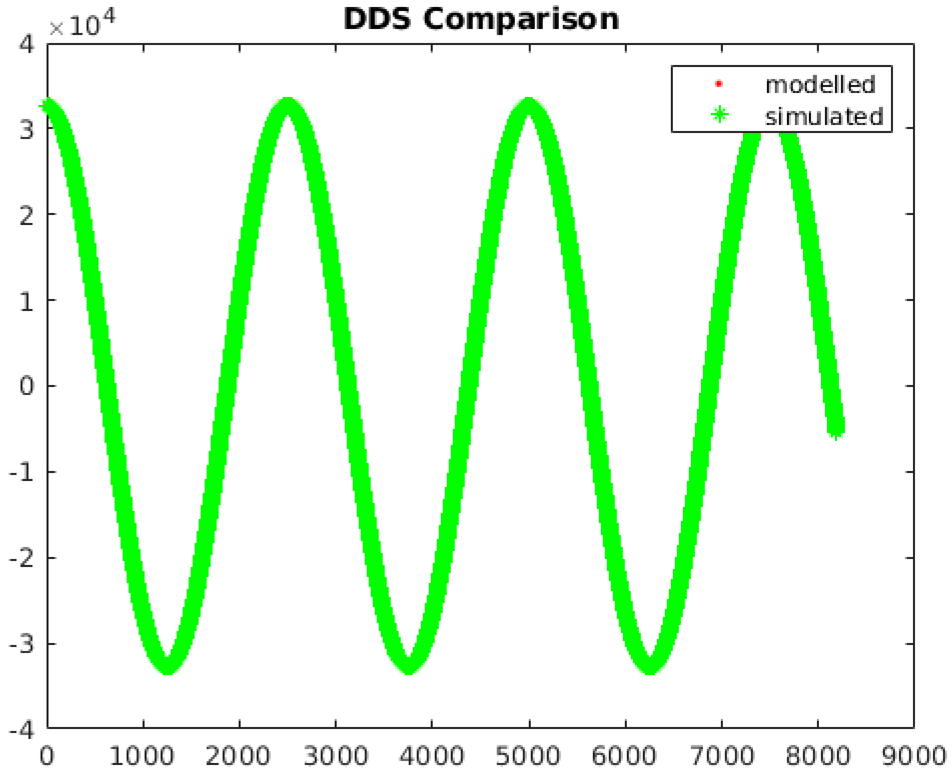
```
%! ----------------------------------------------------------------------
%! SOC Desgin Lab 3
%!  Model dds_compiler IP from Xilinx against bitwise model
%! ----------------------------------------------------------------------

% Varibales
num_points = 8192;
phase_inc = 53687;
sim_filename='simulated_sig_gen.txt';
%ila_filename='ila_100kHz_107374.csv';
read_and_plot_ila_data = false;

% creating DDS with Streaming Phase increment, in Hardware Parameter mode
% (so we can specify the parameters like phase width, output width..etc.)
dds_inst = dds_compiler_v6_0_bitacc('Output_Selection', 1, ...
    'Phase_Width', 27, 'Phase_Increment',3, 'Noise_Shaping', 2, ...
    'DDS_Clock_Rate', 125, 'Output_Width',16, ...
    'Frequency_Resolution', .93132, 'Spurious_Free_Dynamic_Range', 90, ...
    'ParameterEntry', 1, 'Amplitude_Mode', 0);

% Print the config
config  = get_configuration(dds_inst);

% creates an input vector of numpoints length, all are the fixed phase
% increment
data_in = [ones(1,num_points)*phase_inc];

% run the DDS model, which produces the outputs of the DDS
data_out = run(dds_inst,num_points,data_in);

% plot the results
dout_sin = data_out(:,1,2);
plot(dout_sin(1:num_points),'r.');

% load the file from simulation and then plot it on top
simulated_sig_gen = load(sim_filename); %creates simulated_sig_gen
simulated_sig_gen = simulated_sig_gen(1:num_points);
hold on;
plot(simulated_sig_gen,'g*');
title('DDS Comparison');
legend('modelled','simulated');

% this section is relevant to Lab3B only, and won't be run until you
% change the value of "read_and_plot_ila_data" to true
% this part is just showing me parsing the CSV file which I captured
% from my ILA.  The columns in the CSV file, and where the first sample is
% will depend on how you setup your trigger of course, but this is a good
% starting point!  My format is as below...
%Sample in Buffer,Sample in
 Window,TRIGGER,axi_stream_data_from_dds[15:0],dds_valid,iic_sda_i,iic_sda_o,iic_sda_t,iic
%Radix - UNSIGNED,UNSIGNED,UNSIGNED,SIGNED,HEX,HEX,HEX,HEX,HEX,HEX,HEX
if (read_and_plot_ila_data)
```

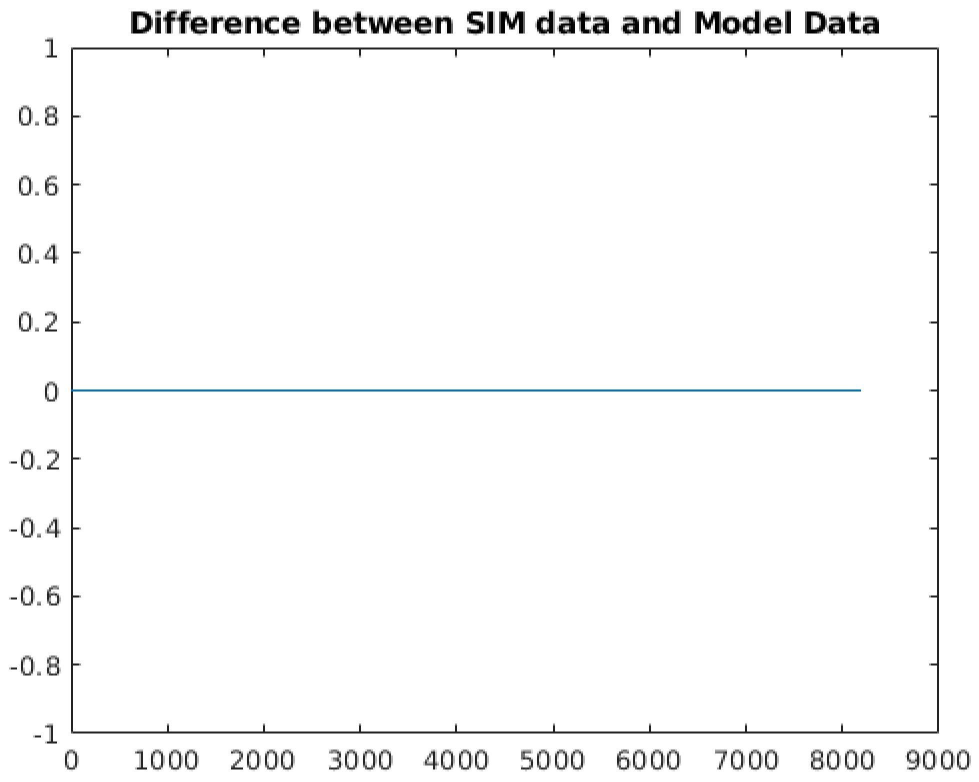
```
    fid = fopen(ila_filename);
    % the textscan is looking for the following fields, which matches the
    % format of my file
    C = textscan(fid, '%d%d%d%d%d%d%d%d%d%d%d', 'headerlines',
 2, 'delimiter', ',');
    fclose(fid);
    sig = C;
    sample_buff = C{1};
    sample_win  = C{2};
    trigger = C{3};
    recorded_sig_gen = double(C{4});
    plot(recorded_sig_gen(1:num_points),'bo');
    legend('modelled','simulated','recorded');
end



% now, plot the difference between the two.  We are striving for 0 here of
% course, but since I didn't simulate the same thing that I modelled, the
% result is not so good.  Note - if you want to make your model match what
% the example simulation file is, it was done with a 27 bit phase
% accumulator, in unit circle amplitude mode, and the phase increment was
% 107374, which is freq = 107374*125MHz/2^27 = pretty close to 100kHz

figure();
plot(simulated_sig_gen - dout_sin);
title('Difference between SIM data and Model Data');



% at this point, I have a signal "dout_sin" which is the output from my bit
% accurate model, and I have "simulated_sig_gen" which is the simulated
% signal that I got from my VHDL testbench, and I have "recorded_sig_gen"
% which is what I captured from the ILA.  All 3 should match perfectly!
% in this case, since I haven't constructed the bit accurate model
% properly, the recorded and the simulated match each other, but not the
% bit accurate model.



disp(['Question 1: When using the unit circle setting it cuts the ' ...
    'amplitude in half. Im not completley sure why, but I think it has '...
    'to do with the hypotenuse of the triangle being unit 1'])

disp(['Question 2: When using Taylor Series Expantion, the DSP' ...
    'utilization is 2 and the BRAM usage is 1. When using no noise' ...
    'shaping there are 0 DSP slices and 15 BRAM slices.'])

disp(['Question 3: My phase inc was 53687. This would create a cosine' ...
    'wave at 49999.915Hz.'])
```

```
Question 1: When using the unit circle setting it cuts the amplitude in half.
 Im not completley sure why, but I think it has to do with the hypotenuse of
 the triangle being unit 1
```

Question 2: When using Taylor Series Expantion, the DSPutilization is 2 and the BRAM usage is 1. When using no noiseshaping there are 0 DSP slices and 15 BRAM slices.

Question 3: My phase inc was 53687. This would create a cosinewave at 49999.915Hz.





Published with MATLAB® R2023a