PreLab 7: AC Gain-Bandwidth Test

In this lab, we will design a test plan for the AC Gain-Bandwidth. This lab is intended to teach you how to map the sampling theory discussed in class onto real equipment. Most of the code will be written as a group in lab. However, to get you ready and thinking about what we will do, please perform the following functions for your prelab.

a) Our DIB cannot be placed in the unity-gain bandwidth configuration. To compensate, we are going to utilize the gain-bandwidth trade-off. In previous courses, we proved that if an op-amp is placed in the standard noninverting op-amp configuration, the gain-bandwidth trade-off holds (e.g. Av*f3dB=GBW). Print out a copy of the DIB schematic and show the paths necessary to get the device into this configuration. Chose a path that will minimize the feedback resistance (the GBW is pretty small for this amplifier, so we don't want to bring the f3dB down too far). What is the resulting RF value? R1 value? The input should use an APU that contains an AWG to generate the AC waveform. The output should use a QMS so that an FFT can be performed (the QMS is the only resource that we have that contains on on-board DSP chip).

b) Will you need to add any more instrument groups or #defines? Explain your answer.

c) You will generate a two-tone signal to be used in the GBW test. One tone will be the f3dB given the feedback that you found in part (a). This signal should be fairly accurate. The second tone should be well in the passband. Although an exact frequency is not required, the final frequency should be indicated. Assume that we will use a 50MHz master clock and a sample size of 256 elements for a single cycle of the coherent waveform. Calculate the divide ratio of the master clock to get the sampling rate (m), the fundamental frequency for both the AWG and the digitizer, and the bin numbers (M) for the two frequencies. Play with m and M to get the an accurate f3dB.

d) Using Matlab, sum the discrete representations of your 2 signals. Make sure that both signals have the same amplitude and that the amplitude will not rail the device. No phase shifting is necessary. Make an array containing 1 cycle of your summed coherent waveform. (Hint: use the time vector $t=0:1/f_{sample}:1/f_{fund}$ to achieve the final array of 257 elements. When we transfer the data to the ETS software, we will remove the final element to create a 256 element waveform, which makes N a power of 2.) Send the final array of discrete signal values to a .csv file that will be used in class.

Lab 7 - AC GBW Test

In this lab, you will add the AC GBW test.

A. UserInit

In this lab, we will use the AWG on an APU board as the input and digitize the output using a QMS.

1. group instruments

We must first set-up the APU_AC_Vin group, the QMS_Vo group, and add the QMS_Vo group.

2. Clocking Scheme

Next, in UserInit, we must set up the clocking scheme and load the input waveform into the AWG. See the prevoius lab if you don't remember how to do this.

3. Loading the waveform into the AWG

Next, load the waveform into the AWG. Create a pattern the same size as your Matlab array. Then, replace the pattern in the "name-awg.h" file with the data from your .csv file. Refer to the previous lab if you don't remember how to do this.

B. Create the Test Function

Create the AC function as you have in the past. Set up the datasheet file to look for $AV_{3dB}/AV_{midband}$. This value should be greater than or equal to 0.707 to meet the 3dB requirement. Since the APU is generating the input signal, a DSP processor is not available so an FFT cannot be taken on the input. However, if you created your input waveform in matlab so that both tones have the same amplitude, the FFT of both input tones should be the same. Therefore, when you calculate the $AV_{3dB}/AV_{midband}$, the input should cancel out, leaving you with $Vo_{3dB}/Vo_{midband}$.

Close the relays and set up the rails if you need to. Next, the APU and QMS must be set-up.

1. Set the apu to force the AWG pattern and select the awg pattern you want to use.
2. Set up the QMS to digitize.
3. Connect the clocks and run the sequence.
4. Take the FFT using the on-board dsp. The spectrum function takes the FFT and plots only the positive frequency components. The fft function produces an FFT with positive and negative frequency components. Use the spectrum function.
5. Get the data from the transformed array. You want the data at the 3dB frequency and at the midband frequency. Get the magnitude of the transformed data using a peak function around the 3dB frequency and midband frequency. The spectral bin (the M value in your calculations) corresponds to element number in the array Set start field to the M value minus 2 and the stop field to the M value plus 2. Your data should be contained in this range.
6. Store the results in a results structure.
7. Perform math on the function and log the result.
8. Turn off all equipment and open the relays.
9. Don't forget to modify the TestCompletion and FailSite if necessary.

C. Test the code

First, verify that the code is working as expected.

1. Compile and build the program. Fix any errors.
2. Look at both the input and the output. Make sure that the output is not clipping.
3. View the FFT. Make sure that the primary peaks are at the expected frequencies.

4. Make sure your function datalogs correctly.
5. Repeat 2, 3, and 4 on the ATE.

D. Write up Your Results

Record your test results and your test time. Be sure to add a picture of your FFT. Turn in your final code, test results, and a discussion of the results.