PreLab 6 – Pattern-Based Testing

In this lab, we will test the LM741 for Vos, CMRR, and PSRR using 1 test. The rails are the only thing that changes between these tests. If we can change them using a pattern, we can potentially speed up the test time. An important note, however, is that while pattern based testing can improve speed, it is at a cost of flexibility.

a) Print out the schematic of the DIB and highlight the pathways necessary to test Vos, CMRR, and PSRR using the nulling amp.
b) I was not able to add the GoI test to the pattern. Why not? (Hint: You cannot switch relays during pattern based testing. You can only change voltages on resources that have an AWG connected to them.)
c) What voltages should be used to test Vos, CMRR, and PSRR?
d) You should find that the only voltages that change for each of these tests are the supply voltages. Sketch VCC and VEE as a function of time. Each test should be completed in 1ms.
e) The AWG will be driven with a clock that is operating at 100kHz. To maintain a constant voltage, the clock must source multiple samples of the same value. How many samples must you have for each test?

Lab 6 - Pattern-Based Testing

In this lab, you will add the pattern-based test to find Vos, CMRR, and PSRR.

A. UserInit

In this lab, we will use the AWGs on the SPU boards for the supplies and digitize the output of the nulling amp using a QMS. Since these resources have already been set up, no instrument groups must be added.

In UserInit, we must set up the clocking scheme and load the input waveform into the AWG. This “one-time set-up” should only be done once per lot, so it should be placed in UserInit.

Clocking Scheme

The ATE allows for multiple clocks to be set-up at different clock-frequencies to handle mixed-signal test applications. In order to setup the clocks for the AWG and the digitizer, we must determine the master clock frequency and the multiplexing scheme to get the divided down signal to the channel of interest.

1. Set the master clock frequency equal to 50MHz.
2. Divide down the master clock down to 100kHz to drive the AWGs.
3. Connect the clocks to the AWGs.
4. Load the waveforms into the AWGs.
5. Set the number of clocks necessary to drive the AWGs.

6. Disconnect the clocks from the AWGs so the clock does not add noise during high-precision DC measurements.

B. Create the Test Function

Create the Test_Pattern function as you have in the past. You will need 3 rows in the datasheet since you are datalogging 3 parameters: Vos, CMRR, and PSRR.

1. Close the relays necessary.
2. Set the APU in AWG mode.
3. Select the AWG pattern.
4. Set the QMS in digitize mode.
5. Reconnect the clocks to the AWGs and QMS.
6. Run the clocks. Run through the sequence 1 time.
7. Average multiple measurements to remove noise.
8. Store the results in a results structure, or you will lose your data.
9. Perform any math necessary and store the results in a double array and datalog.
10. Repeat steps 7-9 for all of the specs that you are measuring.
11. Turn off all clocks, equipment, and open the relays.
12. 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 the patterns in the AWGs and make sure they are correct.
3. Make sure your function datalogs correctly.
4. Repeat 2 and 3 on the ATE. You may need to adjust your number of points and/or clock frequency to give the circuit enough time to stop ringing and get a good measurement. The nice thing is this shows you exactly what the output is doing.

D. Write up Your Results

Record your test results and your test time. Be sure to include pictures of your waveforms. Turn in your final code, test results, and a discussion of the results.