# PreLab 6: DAC Zero Offset and Gain Error

In this lab, we will design the function to test the zero offset and gain error specifications of the TLV5616 DAC.

a) Draw a schematic and describe the test procedure to test the zero-scale-error and gain error. Note: zero-scale-error must be measured at code 10 (look at Fig 19 on page 18). Explain why this is the case.
b) Find the calculation and datasheet limits for these 2 specifications.
c) Look at Fig 1 on page 6 and the digital timing requirements on page 5 to determine the clocking requirements of the DAC. To meet the minimum pulse duration for SCLK high and SCLK low, what is the fastest clock frequency and what duty cycle should the clock have?
d) Is the frame select (FS) positive or negative edge triggered? What is the minimum time between the negative edge of FS and the negative edge of the clock?
e) If you can run SCLK at 50% duty cycle at 20MHz, and you have control over the rising and falling edges of SCLK, FS, and DIN, where should you place the rising and falling edges in the 50ns window to meet all the timing requirements (primarily $t_{wh(FS)}$, $t_{su(FS-CL)}$, $t_{su(D)}$, $t_{h(D)}$) to load the data properly? Assume CS is always low.
f) Look at the DIB schematic. Which relays should be set to get a 10k load resistor adn the reference voltage directly set by an APU? Which CBITs do these relays correspond to?

# Lab 6 - DAC Zero Offset and Gain Error

In this lab, you will develop the routines necessary to test the Zero Offset and Gain Error specifications.

### A. DPU Clocking

In this lab you will need to use a digital clock to get data into the DAC through the serial port and an analog clock to digitize the output of the DAC. In order to do this, you will need to:

a) Divide down the master clock appropriately to drive the master clock and the analog clock.
b) Set-up the digital clock format and timing.
c) Define the analog values that correlate to a digital 1 and 0.

### B. Create the Zero-scale Error (Ezs) and Gain Error (Eg) Function

#### 1. Clock Set-Up in UserInit

For this function, we will be setting up two clocking schemes: a digital clock and an analog clock. The recommended operating conditions list a 20MHz digital clock. The analog QMS will be used to measure the output. The maximum frequency for the QMS using the 20V range is 200kHz using the 200kHz filter. Therefore, we will drive the QMS at 100kHz so that the data won't be attenuated by the filter to receive more accurate measurements.

Analog Clock Set-Up
(i) Set the master clock to 40MHz so it can divided down to both clocks with an integer
(ii) Divide the 40MHz down to 100kHz

Digital Clock Set-Up
(i) Divide the master clock down to 20MHz.
(ii) Set all pins in the digital vector file to input voltage 5 and 0 volts and a receive voltage of 4 and 1V.

Digital pin format*:*
(i) The clock should have the clock format (KT), so that each digital clock will have a rising and falling edge. Set the edges at 20 and 70 percent of the full cycle.
(ii) The frame select (FS) pin can have a return to zero format (RZ) so that it will remain at zero until activated by a "1". Set the edges at 30 to 80 percent so the negative edge will occur after the negative clock edge.
(iii) Din and CS can have no format (NF) since CS will always be tied low and Din will be set by the level.

**2. Load the Digital Inputs using the Vector Editor**

The goal of this test is to time the analog and digital clocks so that they are synchronized. Run the digital vectors to load the serial data and latch it into the ADC. Make the digital clock waits until the analog clock can be synchronized and Vo changes values. Then, make the digital clock wait while the analog clock digitizes the results using the QMS.

(a) Create a new sheet in the Vector Editor with a different label.
(b) Add Digital Pin Groups to this Page
(c) Set up the Input Vectors: For each digital code, we will need to set the frame and load the data serially (17 digital cycles = 425ns), and then give the data time to settle (5.5us). This comes to a total of 5.925usec. However, the analog clock is 10usec. To synchronize the analog and digital clocks, we need to load the data and then wait the duration of the 10usec for the analog clock to finish its "off" cycle. Then we need to wait an additional 10usec for the analog clock to measure and store the results. Set up the inputs to make this happen.

(i) Insert 39 steps into the vector editor using "Insert -> steps" and enter 39.
(ii) On the first step (which should also contain the digital label), set up the digital inputs (called vectors) so that the frame select (FS) is high to start latching the data. Since this is in "return to zero" format, set this to a 1 to force it high. CS should be low at all times. Since the clock is in "clock" mode, a 0 or a 1 will force a clock pulse. You can leave this as a 0. Since Din is in no format, this must be set to the level you want it to be. Set this as a zero initially.

(iii) On the second step, start serially loading in the data. The MSB of Din is a don't care. Set this to anything. Make sure that FS is low so another frame is not started.

(iv) On the third step, continue loading in the data. Set Din=1 to set the DAC in "fast" mode.

(v) On the forth step, set Din=0 to set the DAC in "full power" mode.

(vi) On the fifth step, the Din is a don't care.

(vii) On the sixth step, set Din=0 to load the MSB of the digital value. To make the offset voltage measurement, you will be loading a binary 9 to get 10mV on the output. This will place the first measurement out of the dead zone, so that an accurate offset measurement can be obtained.

(viii) Continue entering Din bits until the 12 bit binary 9 is loaded.

(ix) On the eighteenth step, hold the data and wait for the D/A to settle and the analog clock to synchronize. Wait 183 cycles to synchronize the analog and digital clocks. (There are 200 20MHz cycles in 1 analog clock (10usec). We used 17 cycles to load the data. Wait the remaining 183 to synchronize to the analog clock.)

(x) On the ninteenth step, hold the data for another 5 analog cycles so that 5 measurements can be made on the slow analog system and averaged later (this equates to 1000 steps).

(xi) Repeat this for the full scale code. A total of 10 analog measurements should be made.

(xii) Go back now and modify the analog clock sequence. To map the analog and digital clocks together, you want the analog sequence to be off 1 cycle (wait for digital load), on 5 cycles (get 5 measurements) for both parameters.

## 3. Create the Zero Offset / Gain Error Function

(a) Create a function called EzsEg using the Wizard.

(b) Set up the Datasheet file to datalog zero-scale error and gain error.

(c) Set up the analog resources (assume ground will be connected to the DIB ground). The digital resources are set-up entirely in the digital vector editor.

(d) Set up the QMS to digitize the results.

(e) Connect the QMS to the clock.

## 4. Run Both Clocks and Store the Results

(a) Go back to UserInit and set up the analog clock to drive off 1 clock cycle and on 5 cycles to obtain the 5 analog measurements. Repeat it for the second digital input.

(b) Back in the function, use the run the clocks to drive the analog and digital resources.

(c) Transfer the data.

(d) As an interim test of the vector editor, compile and build the program and correct any errors.

(e) Run the program with a break point at the clock run command. At the break point, display the clocks and verify that your clocking scheme is correct, your clocks are the correct frequency, and that the QMS is taking 10 samples.

(d) Back in you function, find the average of each measurement and store the results in a results structure.

(e) Calculate the offset voltage and gain error as indicated in the datasheet.

(f) Datalog the results.

### 5. Validate your Program

(a) Make sure that your program compiles and runs without alarms on your laptop.

(b) Backup your program and move it over to the ATE.

(c) Run your test on the DAC chips. The necessary wait time to move a digital input from 10 to full-scale is not given on the datasheet. Therefore, you will probably need to adjust the wait time for the circuit to settle. To do this you must (i) increase the wait time in the vector editor and (ii) increase the number of "OFF" cycles in the analog clock. Be sure to change these in a manner that maintains clock timing.

(d) Be sure to include a copy of the QMS output in addition to the datalog and timing information in your final report.

(e) Repeatability study: I would like for you to perform a repeatability study using this function. To do this, you will need to run your test program on one DAC chip 25 times. Save the datalog file. Use the Matlab script ("Repeatbility") to analyze your data. This script will create a histogram and calculate Cp, Cpk, and GRR for both of your functions.

## C. Write up Your Results

Turn in your final code, datalog results, pictures of your QMS data, and results of repeatability study. Discuss your repeatability results in your lab memo and how you could improve the results if necessary.