Name: \_\_Replace with your name(s)\_\_\_\_

EID: \_\_Replace with your EID(s)\_\_\_\_\_

Semester: Fall 2024

Course: ECE445L

A) ***Objectives*:**

1. Describe in a few sentences what the purpose of this lab is.

B) ***Hardware Design Deliverables:***

1. Using KiCad or pen & paper, draw the circuit used to produce the analog voltage you measured. For most students, this is simply which of the two circuits in Figure 2.1 you used.

C) ***Software Design Deliverables:***

1. I have pushed my code to Git Hub for grading (Check box if true).

D) ***Measurement Data:***

1. **Measuring noise with the DVM**.

Signal =

Noise =

SNR =

ENOB =

1. **Measuring noise with the oscilloscope**.

Signal =

Noise =

SNR =

ENOB =

1. **Observing the noise profile with a spectrum analyzer**. Show photo or screen shot of amplitude versus frequency.
2. **Measuring noise using software**.

Signal =

Noise =

SNR =

ENOB =

1. **Measuring time jitter.**

Tmax =

Tmin =

Jitter =

1. **Observe the critical section.** Take a screenshot or photo like Figure 2.4.
2. **Explore the behavior of the CLT.**

|  |  |  |  |
| --- | --- | --- | --- |
| Averaging | SAC | SNR | ENOB |
| 1 | **0** |  |  |
| 2 | **1** |  |  |
| 4 | **2** |  |  |
| 8 | **3** |  |  |
| 16 | **4** |  |  |
| 32 | **5** |  |  |
| 64 | **6** |  |  |

Two PMF plots: one for SAC = 0 and another with your choice of SAC, like Figure 2.5.

1. **Perform a calibration**: show a table with truth, ADC, and measured value. Calculate the average error of the calibration.

E) ***Analysis and Discussion Questions*** (1 or 2 sentences each):

1. This system toggles GPIO pins to profile when and where the software is executing. Is this debugging intrusive, nonintrusive or minimally intrusive? Justify your answer.
2. Finish these 4 statements.

The advantage of the DVM to measure noise is

The advantage of the scope to measure noise is

The advantage of the spectrum analyzer to measure noise is

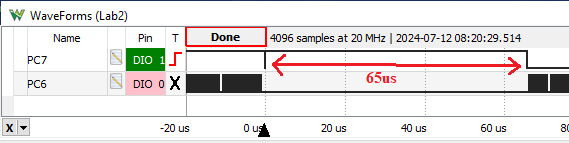
The advantage of the software to measure noise is

1. Explain the method you used to remove the critical section. What other approaches could you have used?
2. This line of code involves a divide operation.

**InvokesDivide = (InvokesDivide\*12345678)/12345;**

The divide instruction on the Cortex M4 requires 2 to 10 bus cycles to execute, and it is atomic. Explain why this line of C does **not** create a 10 bus-cycle sampling jitter. *Hint*: observe the assembly code created by the compiler. *Note*: the answer will depend on the compiler. I.e., your compiler might create a 10 bus-cycle jitter.

1. The PMF results will show hardware averaging is less noisy than not averaging. If it is so good, why don’t we always use it? Compare the following logic analyzer trace to Figure 2.4. Explain why this configuration required 65us to execute the ISR.

******

1. **ADC Resolution** is defined as the change in analog input that can be reliably detected. A simple estimate of ADC resolution is the standard deviation your software calculated when the input is constant. Another estimate of ADC resolution uses the PMF. The following four PMFs were obtained with a precision analog reference voltage connected to the analog input of a TM4C123 LaunchPad. Hardware averaging of 64 was used. Using the PMF plots, if the input were to change from 1.6500V to 1.6505V it could not be reliably detected, because the PMFs overlap. However, if the input changes from 1.6500V to 1.6510V, it can be reliably detected, if we define reliably as “more likely than not”. Therefore, what is this ADC resolution?

