ECE445L Spring 2025 Quiz 2 EID = _____

Name:

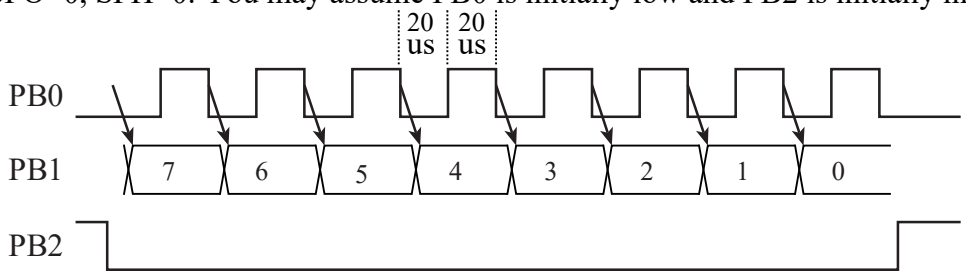
April 3, 2025, 12:30 to 1:45 pm. Closed book, one page of crib notes. No electronics devices.

**(10) Question 1.** Sensor output is $0<(V_1-V_2)<0.1V$. The ADC input is $0<V_{out}<3V$. Build an interface between the sensor and the ADC. Show chip numbers and resistor values. The system needs high input impedance, low output impedance, and good CMRR.




**(5) Question 2.** See the data sheet for the MAX5353 DAC on the last page. Write a timing equation for **data required interval** for this DAC. The interval is in the format **(xxx, yyy)**.

**(15) Question 3.** Your microcontroller is the SPI master sending data at 25 kbits/sec. One 8-bit data transmission requires about 8*40=320 us to complete. These three pins are not supported by hardware SPI in the TM4C123, so you will have to implement the interface using simple GPIO output. The following timing defines SPO=0, SPH=0. You may assume PB0 is initially low and PB2 is initially high.




Consider these nine possible actions.

**W20** means wait 20 us
**W40** means wait 40 us
**PB0=1** means set PB0 high
**PB0=0** means clear PB0 low
**PB1=2** means set PB1 high
**PB1=0** means clear PB1 low
**PB2=4** means set PB2 high
**PB2=0** means clear PB2 low
**Out** means output one valid data bit to PB1

**(3) Part a)** Which action occurs once and first?

**(10) Part b)** List the actions in proper order that transmit one bit. This sequence will occur 8 times. Make sure you maximize the data available interval so the slave can receive the data properly.

**(2) Part c)** Which action occurs once and last?

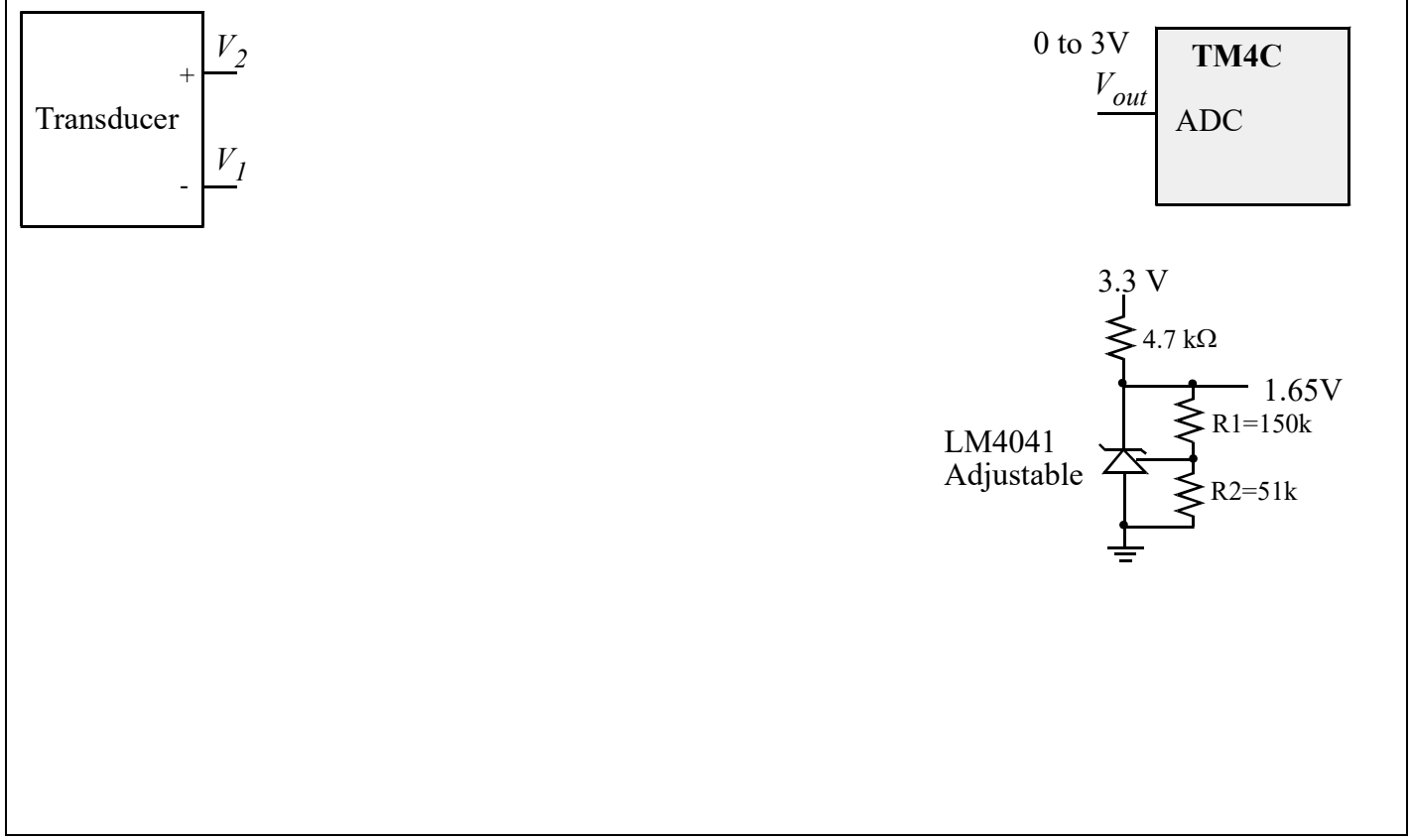
**(5) Question 4.** You will use **binary fixed-point** to implement *area* equals *width* times *length*. Assume *width* and *length* are fixed-point numbers with 1/16 cm resolution; **`W`** and **`L`** are the unsigned integer parts respectively. Assume *area* is a fixed-point number with 1/16 $cm^2$ resolution; **`A`** is the unsigned integer part of *area*. **Write C code** that calculates **`A`** as a function of **`W`** and **`L`**.

```
uint32_t W,L,A;
```

**(15) Question 5.** Design an analog circuit with the following transfer function: $V_{out} = 2*V_1 - V_2$. You may assume inputs $V_1$ and $V_2$ are such that the range of $V_{out}$ will be 0 to 3 V. Show your work and label all resistor values. You do not have to show pin numbers. You do not need to add a low pass filter. Use E24 values

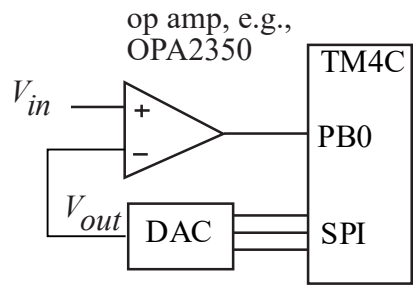
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 11 | 12 | 13 | 15 | 16 | 18 | 20 | 22 | 24 | 27 | 30 |
| 33 | 36 | 39 | 43 | 47 | 51 | 56 | 62 | 68 | 75 | 82 | 91 |

**Table 9.2. E24 Standard resistor and capacitor values for 5% tolerance.**

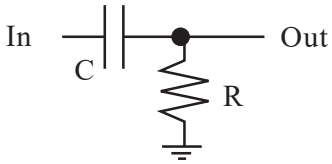


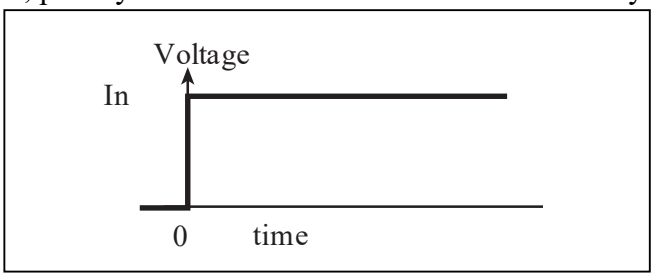
**(15) Question 6.** Write a software function to implement a **successive approximation** ADC. The hardware is given. **PB0** is initialized to an input. Reading **PB0** will return 0 or 1. SPI is initialized to output to an 8-bit DAC. The range of the DAC matches the expected range of the ADC, which is 0 to 3V. To output to the 8-bit DAC, you can call **DAC_Out(n)**, where **n** is 0 to 255. The algorithm must be successive approximation.

```
#define PB0 (*((volatile uint32_t *)0x40005004))
void DAC_Out(uint8_t n);
uint8_t ADC_In(void){
```




**(9) Question 7.** Consider the following *RC* circuit. At time 0, the input, **In**, instantaneously rises from 0 to 3.3V. Recall that **τ**=**R***C. Sketch the response of **Out** as a function of time. Draw **Out** versus **time** on top of the **In** versus **time** plot. I.e., place your answer inside the box. Label **τ** on your sketch.



**(10) Question 8.** PB7 output is connected to a logic analyzer. Assume **PB7** has been initialized as an output.
**(5) Part a)** Add debugging code to this ISR that can be used to measure interrupt period and percentage CPU utilization for this ISR task. To output, write 0x80 or 0 to **PB7**. Reading **PB7** returns 0x80 or 0.

```
#define PB7  (*((volatile uint32_t *)0x40005200))
void SysTick_Handler(void){




   Task();

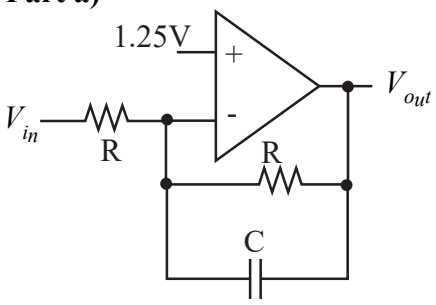



}
```

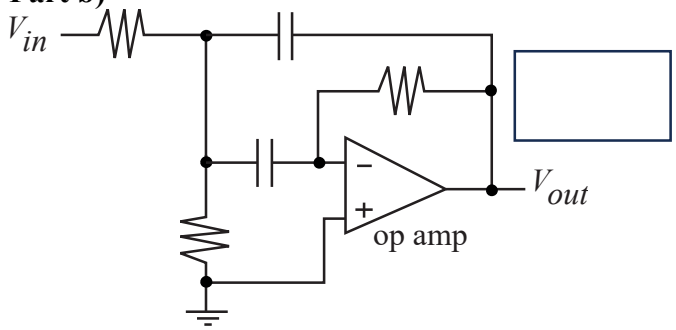
**(5) Part b)** Show the logic analyzer trace for **PB7** that would occur if the interrupt frequency were 10 kHz, and the CPU utilization were 25%. Label the time axis with numbers and units. Show at least two interrupt invocations.

**(6) Question 9.** Classify these analog circuits. Put **LPF HPF** or **BPF** into the boxes

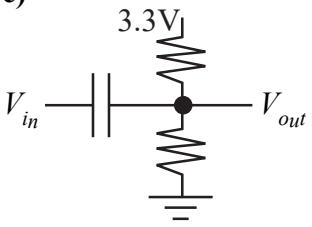
**Part a)**



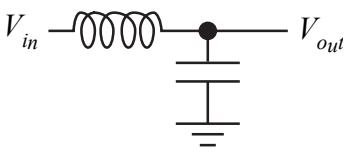

**Part b)**



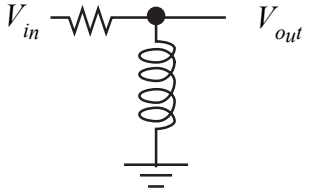

**Part c)**



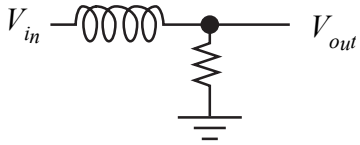

**Part d)**




**Part e)**




**Part f)**

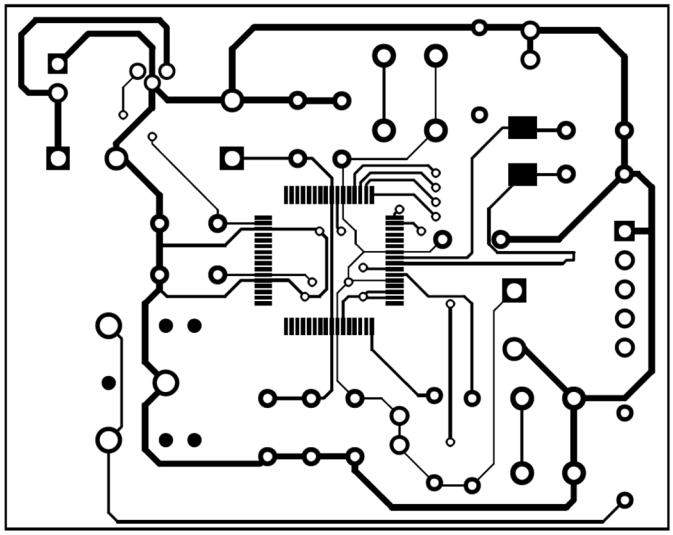



**(10) Question 10.** The following layout shows a simplified Lab 6. The letters and numbers are silk. It is a complete embedded system that will output squarewave sounds on the audio jack based on user input from the switch. There are no errors on the schematic, and it passes DRC with no errors. Unfortunately, there are many PCB layout mistakes. The way this question will be graded is

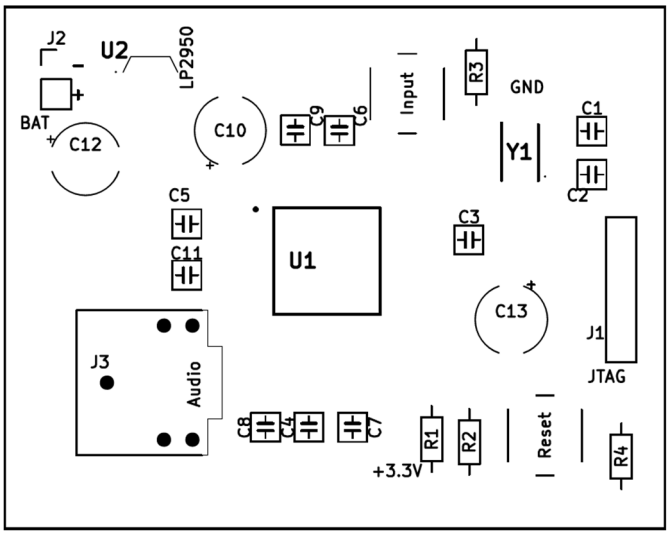
+3 points for each serious layout mistake you identify, i.e., one that causes it not to work
+1 points for each minor mistake you identify, i.e., it works, but the PCB could have been better
-1 points for incorrectly marking something that is not a mistake

For each mistake, circle the place on the "all layers" PCB with a mistake, draw a line to the box below, and explain what the mistake is. The mistakes are all fundamentally different, and not multiple instantiations of the same type of mistake. None of the errors involve the schematic, the silk, component values, or component types.

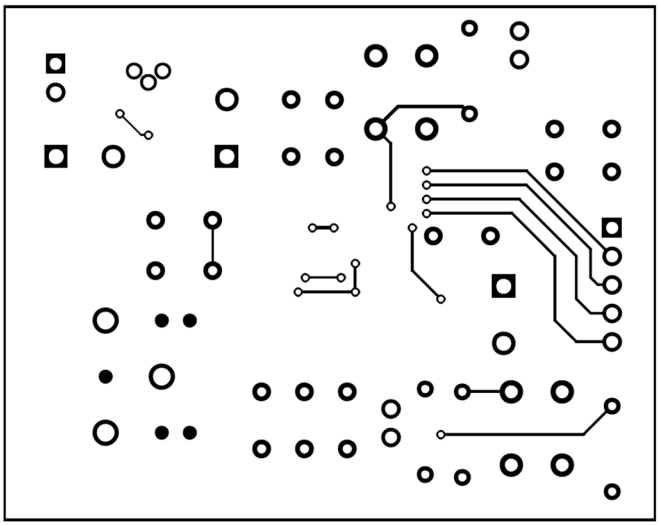
An example minor error is illustrated, "placing capacitors too far from the microcontroller".



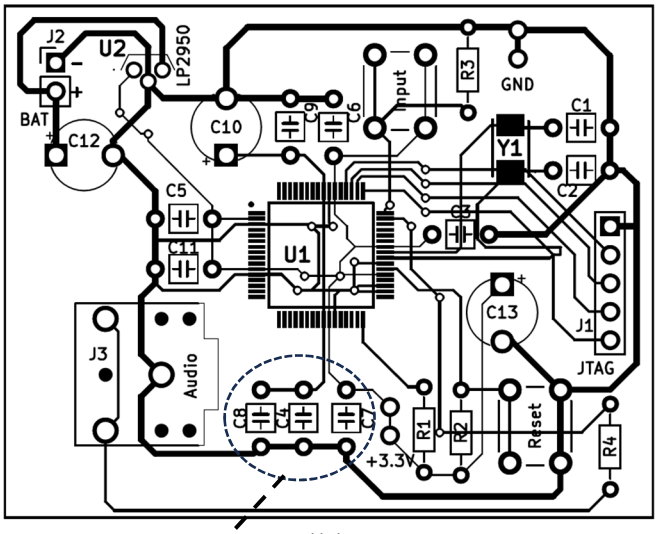
Top copper and edge cut




Top silk and edge cut

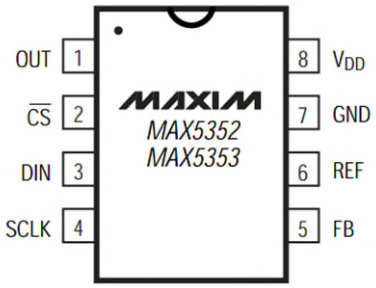


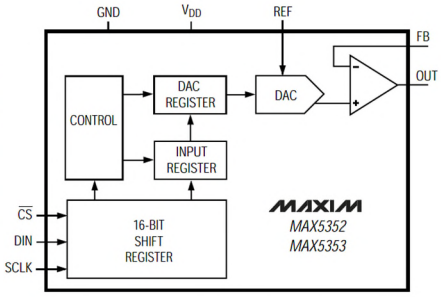
Bottom copper and edge cut

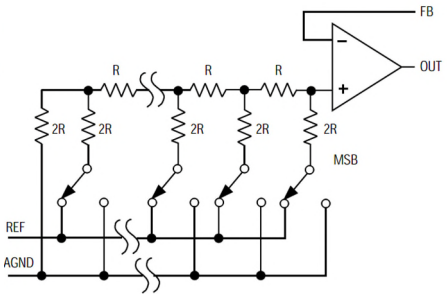



All layers

E.g., C6 C9 C10 C5 C11 C8 C4 C7 C3 could have been closer to U1




**TIMING CHARACTERISTICS** (Figure 6)

| | | | | |
|---|---|---|---|---|
| SCLK Clock Period | $t_{CP}$ | | 100 | ns |
| SCLK Pulse Width High | $t_{CH}$ | | 40 | ns |
| SCLK Pulse Width Low | $t_{CL}$ | | 40 | ns |
| $\overline{CS}$ Fall to SCLK Rise Setup Time | $t_{CSS}$ | | 40 | ns |
| SCLK Rise to $\overline{CS}$ Rise Hold Time | $t_{CSH}$ | | 0 | ns |
| DIN Setup Time | $t_{DS}$ | | 40 | ns |
| DIN Hold Time | $t_{DH}$ | | 0 | ns |
| SCLK Rise to $\overline{CS}$ Fall Delay | $t_{CS0}$ | | 40 | ns |
| $\overline{CS}$ Rise to SCLK Rise Hold Time | $t_{CS1}$ | | 40 | ns |
| $\overline{CS}$ Pulse Width High | $t_{CSW}$ | | 100 | ns |

