November 21, 2014, 10:00-10:50am. Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You have 50 minutes, so please allocate your time accordingly. Please read the entire quiz before starting.

(5) Question 1. What is the difference between a buck-boost and a linear regulator? Pick the answer that best differentiates the two regulator types. Put your answer in the box.

A) A linear regulator needs capacitors on both input and output, but the buck-boost does not need capacitors.

B) The linear regulator only creates an output voltage that is less than the input voltage, and the buck-boost only creates an output voltage that is greater than or equal to the input voltage.

C) A linear regulator can be used to create a power voltage, whereas a buck-boost is used to create a low-noise analog reference voltage for analog circuits.

D) A linear regulator does not exhibit back EMF, but a buck-boost requires a snubber diode because of the inductor in the circuit; the dI/dt in the inductor will cause a large back EMF voltage.

E) The linear regulator is only used for currents less than 1 A, while the buck-boost is only used for currents above 1 A.

F) Assume the current is 1 A, the input voltage is 9 V, and the output voltage is 3.3 V. A linear regulator will get hot and a buck-boost will not get hot.

G) A linear regulator is better for a battery powered application because the large dropout voltage allows the battery to discharge for longer before the battery voltage finally drops out of range.

(10) Question 2. For each application choose busy-wait synchronization or interrupt synchronization. Specify “BW” for busy-wait, and specify “Int” for interrupts. Place your answers in the boxes.

A) With a UART transmission such that packets of 16 or fewer frames are to be sent every 1 second. The baud rate is 115,200 bits/sec. The protocol is 1 start bit, 8 data bits, even parity, and one stop bit.

B) With an SSI interface of a device that requires exactly three frames, two output and one input, and there is no delay between outputs and inputs. The SSI clock is 10 MHz and frame size is 8 bits.

C) With software-start ADC sampling, 1 MHz ADC mode, and no hardware averaging. The sampling rate is aperiodic (not a regular rate).

D) In a PLL initialization where if the PLL does not start, you do not wish to continue execution.

E) An SSI interface between two microcontrollers that wishes to transfer 1000 bytes of data from one microcontroller to the other as quickly as possible in a dedicated fashion.
(15) **Question 3.** The goal is to transmit synchronous serial data as fast as possible using SSI. The external device sends data from the outside world into the microcontroller. The microcontroller is the master, and the external device is a slave. The following figure shows the timing of the external device.

**Part a)** Assume SPH=1; what should SPO be?

**Part b)** The time \( t_i \) is [50, 100ns]. What is the shortest SSI clock period that this device can be interfaced? You may assume S4 and S5 are zero, and the clock will be 50% duty cycle. Show your work.
(10) **Question 4.** Assume GPIO Ports A, B, C and D are already initialized to interrupt on rising edges of PA7 PB7, PC7 and PD7. Also assume interrupts are armed and enabled. **Write C code** to set the priority so that PA7 is the highest, PB7 is the next highest, and PC7/PD7 are equal priority. Assume there are other priority 3 interrupts that are less important than any of these edge-triggered interrupts.

(10) **Question 5.** Design a two-pole Butterworth low pass filter with a cutoff frequency of 51 Hz. Show your work. Specify RA, C1A, and C2A. Get the filter to work; you do not need to specify standard resistor and capacitor values.
(10) **Question 6.** You will use **decimal fixed-point** to implement area equals width times length. Assume width and length are fixed-point numbers with 0.01 cm resolution; \( \text{W} \) and \( \text{L} \) are the integer parts respectively. Assume area is a fixed-point number with 0.01 cm\(^2\) resolution; \( \text{A} \) is the integer part of area. **Write C code** that calculates \( \text{A} \) as a function of \( \text{W} \) and \( \text{L} \).

(15) **Question 7.** Design an analog circuit that maps \(-1 \leq V_{\text{in}} \leq 0.5\text{V}\) into \(0 \leq V_{\text{out}} \leq 3\text{V}\). The input, \( V_{\text{in}} \), is a single voltage (not differential). The output, \( V_{\text{out}} \), is connected to the microcontroller ADC. You may assume the input is bounded between -1 and 0.5V. R1 and R2 are already chosen such that the analog reference is 2.00V. You will use one rail-to-rail op amp. Show your work and label all chip numbers and resistor values. You do not have to show pin numbers.

![Circuit Diagram]
(25) Question 8. The following code uses Timer0A to increment Count on the rising edge of PB6. Edit the code so it uses Timer1A to increment Count on the rising edge of PB4. You can skip the priority register.

```c
volatile uint32_t Count; // incremented on interrupt

void TimerCapture_Init(void){
    SYSCTL_RCGCTIMER_R |= 0x01; // activate timer0
    SYSCTL_RCGCGPIO_R |= 0x00000002; // activate port B
    Count = 0; // allow time to finish activating
    GPIO_PORTB_DEN_R |= 0x40; // enable digital I/O on PB6
    GPIO_PORTB_AFSEL_R |= 0x40; // enable alt funct on PB6
    GPIO_PORTB_PCTL_R = (GPIO_PORTB_PCTL_R&0xF0FFFFFF)+0x07000000;
    TIMER0_CTL_R &= ~0x00000001; // disable timer0A during setup
    TIMER0_CFG_R = 0x00000004; // configure for 16-bit timer mode
    TIMER0_TAMR_R = 0x00000007; // configure for input capture mode
    TIMER0_CTL_R &= ~(0x000C); // TAEVENT is rising edge
    TIMER0_TAILR_R = 0x00000FFF; // start value
    TIMER0_IMR_R |= 0x00000004; // enable capture match interrupt
    TIMER0_ICR_R = 0x00000004; // clear timer0A capture flag
    TIMER0_CTL_R |= 0x00000001; // enable timer0A
    NVIC_PRI4_R = (NVIC_PRI4_R&0x00FFFFFF)|0x40000000; //Timer0A=priority 2
    NVIC_EN0_R = 0x00080000; // enable interrupt 19 in NVIC
    EnableInterrupts();
}
void Timer0A_Handler(void){
    TIMER0_ICR_R = 0x00000004; // acknowledge timer0A capture match
    Count = Count + 1;
}
```