Jonathan Valvano

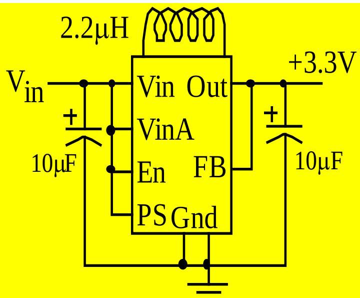
(5) Question 1. Put your answer A,B,C,D,E, or F in the box.

E) TPS63001 3.3V buck-boost regulator

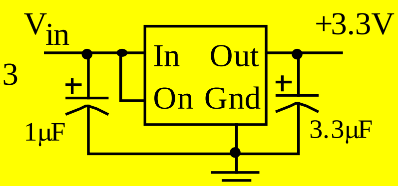
(7) Question 2. Design the circuit

For a linear regulator, we need a dropout less than 0.3V, like LP2981 or TPS78233. You can also use a buck boost like TPS63001

TPS63001



LP2981-3.3



(7) Question 3. Show your equations and the final calculation.

Calculate battery currents for each mode

Sleep mode 0.95*7.4V*I_{sleep} = 3.3V*0.1mA, I_{sleep} = 0.0469mA

Active mode 0.95*7.4V*I_{active} = 3.3V*100mA, I_{active} = 46.9mA

Average current = 0.9*0.0469mA+0.1*46.9mA= 4.74mA

Power budget

2200 mA-hour = *T**4.74mA

T = 2200 mA-hour/4.74mA = 464 hours

(5) Question 4. Part a)

Multiply numerator and demoninator by the same constant

y(n) = (100000**x(n)* – 12558**x(n*-1) +100000**x(n*-2) + 11302**y(n*-1) -81000**y(n*-2))/ 100000

(15) Question 4. Part b) Show the SysTick ISR

```
int32_t X[3],Y[3];
void SysTick_Handler(void){
   X[2] = X[1]; // shift data in buffer
   X[1] = X[0];
   Y[2] = Y[1];
   Y[1] = Y[0];
   X[0] = ADC_In(); // put new data in
   Y[0] = (100000*X[0] – 12558*X[1] +100000*X[2]
       + 11302*Y[1] -81000*Y[2])/ 100000;
}
```

(12) Question 5

(3) Part a) What value did the software write to **DSS** during initialization?

7 (8-bit)

(3) Part b) What value did the software write to **SPO** during initialization?

1 (clock high)

(3) Part c) What value did the software write to **SPH** during initialization?

0 (out on rising)

(3) Part d) What data value is being transmitted (in hexadecimal)?

0x36 (latch on falling)

(10) Question 6. 12-bit ADC measures a 20 kHz sine wave.

(2) Part a) What is the large component at f= 0Hz?

DC offset. Since the ADC is 0 to 3.3V, there must be a DC component

(4) Part b) What is the signal to noise ratio in dB?

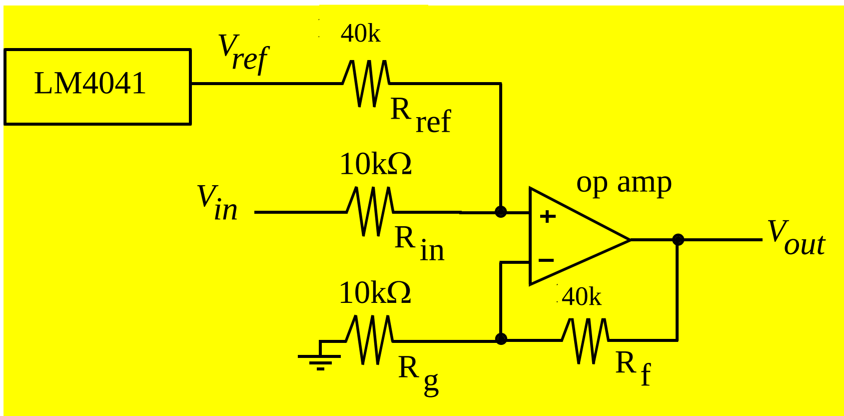
-15 - -55dB = 40 dB

(4) Part c) What is the equivalent precision in bits?

40db =20 $\log_{10}$ N, N (alternative) = $10^{40/20}$ = 100
Precision (bits) = $\log_2 100 = \log_{10} 100/\log_{10} 2$ = 7

(14) Question 7. Design the circuit; show your work

$V_{out} = 4V_{in}+1.5$,
$V_{out} = 4V_{in}+V_{ref}$,
$V_{out} = 4V_{in}+V_{ref} - 4V_g$,
$R_f = 40k$, $R_{in} = 10k$,
$R_{ref} = 40k$, $R_g = 10k$



(10) Question 8 Part a) Show the initialization for Port B edge-triggered interrupts

```
void PortB_Init(void){
```

`SYSCTL_RCGCGPIO_R`	`|= 0x02; delay();`
`GPIO_PORTB_DIR_R`	`=0; // input`
`GPIO_PORTB_DEN_R`	`=0xFF; // enabled`
`GPIO_PORTB_IS_R`	`=0; // edge`
`GPIO_PORTB_IBE_R`	`=0; // one edge`
`GPIO_PORTB_IEV_R`	`=0xFF; // rising`
`GPIO_PORTB_IM_R`	`=0xFF; // all`
	`Max=0;`
`NVIC_EN0_R`	`= 2;`

```
EnableInterrupts(); }
```

(15) Part b) Show the edge-triggered interrupt service routine.

```
void GPIOPortB_Handler(void){
// there is a race condition when reading RIS and DATA
// multiple rising edges can occur simultaneously or near each other
// can't read just RIS, because we need to count old touches too
// eliminate race by reading only one of RIS or DATA
uint32_t data = GPIO_PORTB_DATA_R; // switches that are pressed
uint32_t count = 0; // local or current count
  for(uint32_t mask = 0x80; mask > 0; mask = mask>>1;){
    if(mask & data){
      count++;
    }
  }
  if(count > max){
    max = count; // new max
  }
  GPIO_PORTB_ICR_R = data; // clear only ones we have counted
// rising edges occurring after read data will cause another
}
```