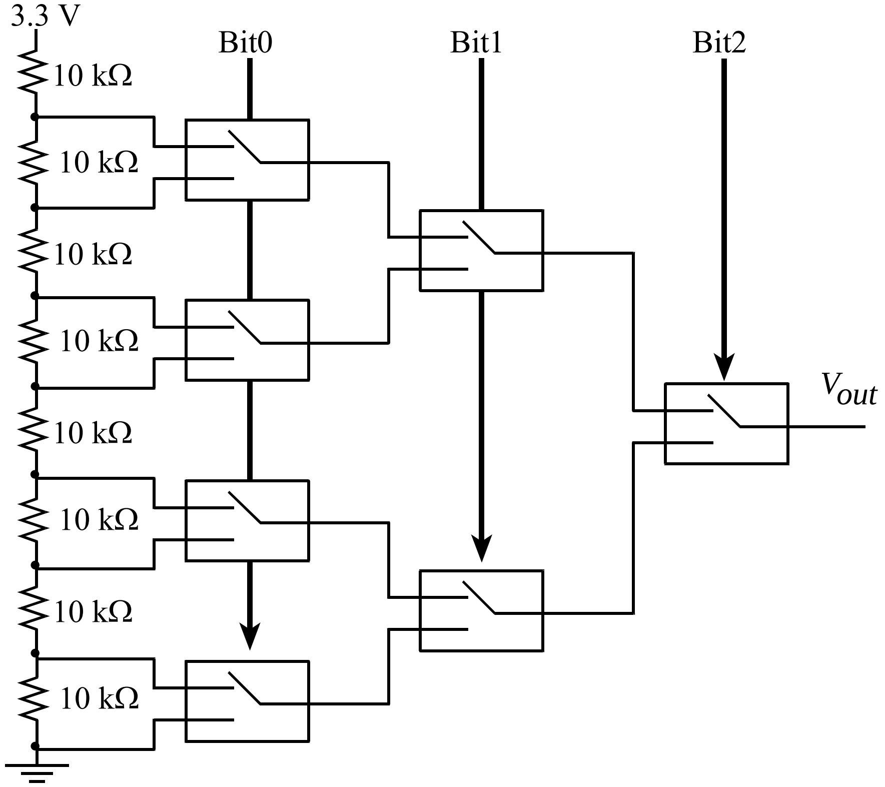
Jonathan W. Valvano

(5) Question 1. For each type of voltage regulator, choose the best description of that device.
(1) Part a) None of the above (regulators are constant voltage, not constant current) You can make a constant current source, but not used as a power source.
(1) Part b) Linear regulator like the LM2937, LP2950, or 78M05. This is why a linear regulator gets hot (I_{in}=I_{out}) so the power dissipated in regulator is (V_{in}-V_{out})* I_{out}
(1) Part c) None of the above (none of these regulators use transformers)
(1) Part d) Boost regulator (increases). The switch and diodes are internal to the device.
(1) Part e) None of the above (almost all regulators require grounds to be connected) An isolation transformer could be used to create power that does not share ground.

(5) Question 2. For each description, choose the best device that matches the description.
(1) Part a) Tantalum capacitors are polarized
(1) Part b) We used two 10 pF C0G ceramic capacitors used in lab for the crystal circuit
(1) Part c) None of the above (a diode used to remove back EMF, the inductor causes the back EMF).
(1) Part d) Resistor, a SSR interface is essentially the same as an LED interface
(1) Part e) None of the above (this is SO weird, nothing behaves like this. Closest is 1/R is I/V.

(10) Question 3. 3-bit resistor-string DAC. Look at how the TLV5616 works, notice the "string of resistors"

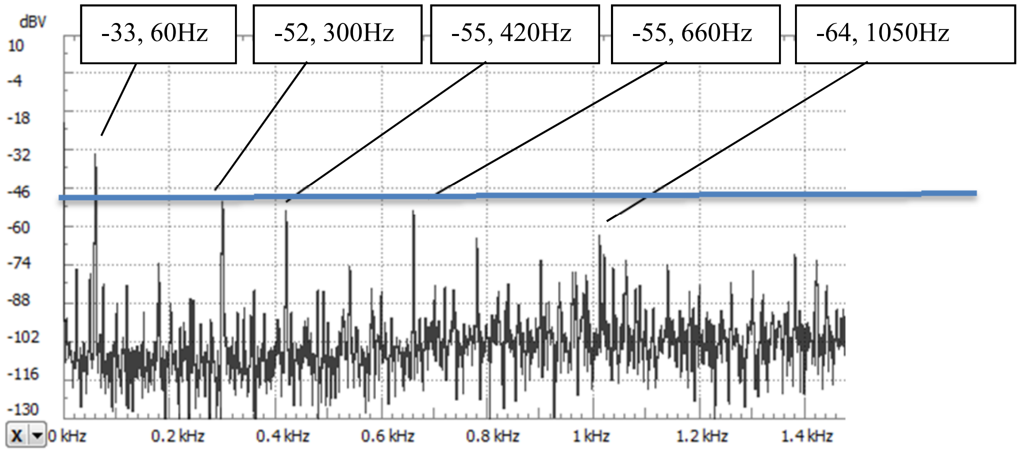


You could also have built it with 7 resistors. Notice an n-bit DAC uses 2^n or (2^n-1) resistors of equal value, and (2^n-1) switches.

(5) Question 4. Civil Rights Act of 1964, IEEE Code of Ethics
IEEE "#8 to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression"
"Outlawed discrimination based on race, color, religion or national origin"

(5) Question 5. coupling.
1) Invocation coupling (one module calls another)
2) Control coupling (action in one module affects the behavior in another). Shared globals fits into this category.
3) Shared I/O devices.

(5) Question 6. This is the output of a typical spectrum analyzer. First we calculate resolution in dB_{FS}



For an 8-bit ADC, the resolution is $dB_{FS} = 20 \log_{10}(2^{-8}) = -48.2$. See Table 10.6 in the book. To prevent aliasing we need input signals about ½ f_s to be less than -48.2 dB_{FS}. To prevent aliasing we could reduce noise, add an analog LPF, or increase the sampling rate (add a digital filter to remove noise). For this question, the sampling rate must be more than 2*60Hz = 120 Hz.

(10) Question 7. The FSM is initialized in main and the FSM controller runs in the SysTick handler.

```
  while(1){
    DisableInterrupts();
    GPIO_PORTF_DATA_R ^= 0x08;
    EnableInterrupts();
    Body();     // other stuff
  }
```

(5) Part a) This is the classic critical section in read-modify-write nonatomic sequence to a shared global, the GPIO data register access in main.
(5) Part b)

1) Disable interrupts in main, during critical section
2) Switch the main program to use a different port
3) Use bit-specific addressing in main
4) Use bit-banding addressing in main
5) Add wait for interrupt in main

(5) Problem 8. 8000 alternatives or 13 bits.

Jonathan W. Valvano First:________________ Last:_____________ EID:__________

Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communi ese pages. Please don't turn in any extra sheets.

(10) Question 9. The following program (Program 6.5) measures the 24-bit pulse width on a signal connected to both **PB6** and **PB7**. Change the software to use **PF0** and **PF1** (the input is connected to both **PF0** and **PF1**). Cross out parts of the code you wish to delete and insert necessary additions.

GPIO_PORTF_LOCK_R = 0x4C4F434B;
GPIO_PORTF_CR_R |= 0x1F;

```
uint32_t PW;                // 24 bits, 12.5 ns units
int Done;                   // set each falling
void PWMeasure2_Init(void){ // TM4C123 code
  SYSCTL_RCGCTIMER_R |= 0x01;      // activate timer0
  SYSCTL_RCGCGPIO_R |= 0x02;       // activate port B
  Done = 0;                        // allow time to finish activating
  GPIO_PORTB_DIR_R &= ~0xC0;       // make PB6, PB7 inputs
  GPIO_PORTB_DEN_R |= 0xC0;        // enable digital PB6, PB7
  GPIO_PORTB_AFSEL_R |= 0xC0;      // enable alt funct on PB6, PB7
  GPIO_PORTB_PCTL_R = (GPIO_PORTB_PCTL_R&0x00FFFFFF)+0x77000000;
  TIMER0_CTL_R &= ~0x00000003;
  TIMER0_CFG_R = 0x00000004;       // configure for 16-bit timer mode
  TIMER0_TAMR_R = 0x00000007;
  TIMER0_CTL_R = (TIMER0_CTL_R&(~0x0C))+0x04; // falling edge
  TIMER0_TAILR_R = 0x0000FFFF;     // start value
  TIMER0_TAPR_R = 0xFF;            // activate prescale, creating 24-bit
  TIMER0_IMR_R |= TIMER_IMR_CAEIM; // enable capture match interrupt
  TIMER0_ICR_R = TIMER_ICR_CAECINT;// clear timer0A capture match flag
  TIMER0_TBMR_R = 0x00000007;
  TIMER0_CTL_R = (TIMER0_CTL_R&(~0x0C00))+0x00; // rising edge
  TIMER0_TBILR_R = 0x0000FFFF;     // start value
  TIMER0_TBPR_R = 0xFF;            // activate prescale, creating 24-bit
  TIMER0_IMR_R &= ~0x700;          // disable all interrupts for timer0B
  TIMER0_CTL_R |= 0x00000003;      // enable timers
  NVIC_PRI4_R = (NVIC_PRI4_R&0x00FFFFFF)|0x40000000; // Timer0=priority 2
  NVIC_EN0_R = 1<<19;              // enable interrupt 19 in NVIC
  EnableInterrupts();}
void Timer0A_Handler(void){
  TIMER0_ICR_R = 0x00000004;// acknowledge timer0A capture flag
  PW = (TIMER0_TBR_R-TIMER0_TAR_R)&0x00FFFFFF;// from rise to fall
  Done = 1;}
```

Change 0x02 to 0x20 to activate Port F

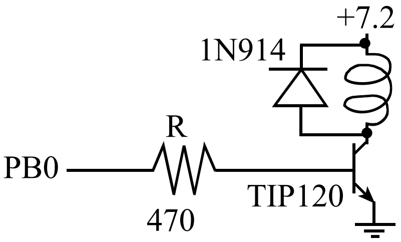
Change all PORTB to PORTF
Change all 0xC0 to 0x03

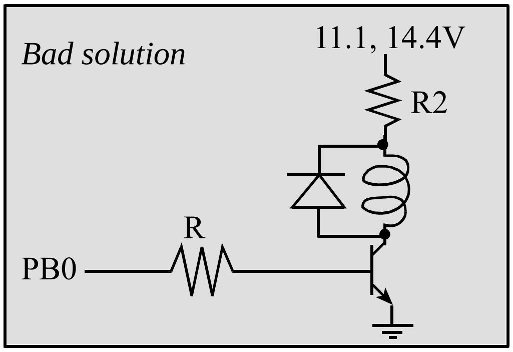
`0xFFFFFF00)+0x00000077`

(10) Problem 10. You will implement a PID controller, assume runs every dt time
Integral looks like I = I+data*dt
Derivative looks like D = (data-old)/dt; then set old=data;

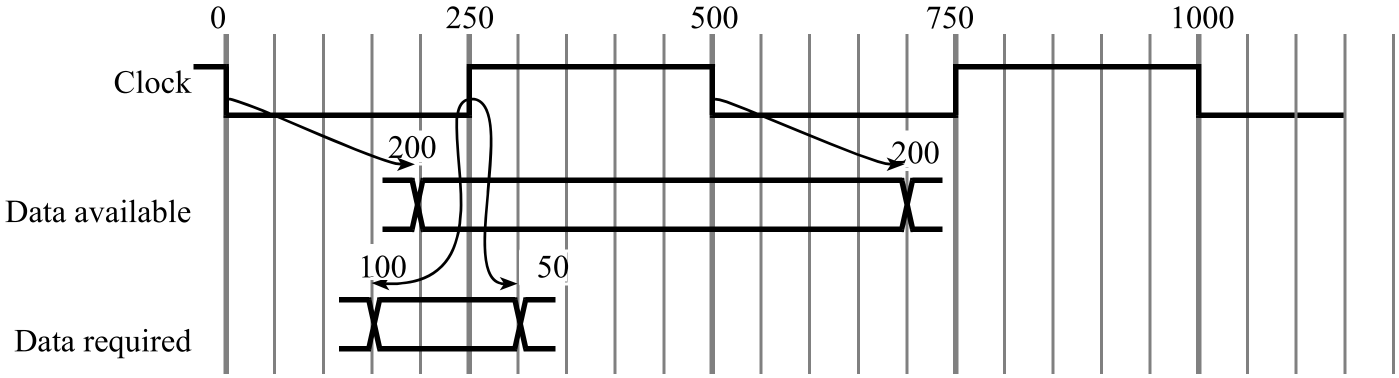
```
void SysTick_Handler(void){ // Executed every 1ms
// Notice that dt is 1ms,
// Do a dimensional analysis to see how simple this problem is
static int16_t Ui,Eold;
 int16_t Up,Ud,U,X,E;
  X = CurrentSpeed();
  E = Xstar-X;       // rps
  Up = E/10;         // proportional Kp*E
  Ui = Ui+E/4;       // integral    (Ki*E)dt
  Ud = (E-Eold)/100; // derivative   d(Kp*E)/dt
  U = Up+Ui+Ud;
  if(U < 0) U=0;   // good idea, but not explicitly required in exam
  if(U > 1000) U=1000;
  SetPower(U);
  Eold = E;  // used for derivative in next interrupt
}
```

(10) Question 11. Interface an electromagnetic relay to the microcontroller.
Choose 7.2 so voltage to motor is 6.7V (7.2-V_{CE}). Choose TIP120 to get I_{CE} of 600 mA. The motor requires 600mA and, the current gain is 1000, so we need 0.6mA of I_B. We guess V_{BE}(on) is about 1 V at I_C=600mA. We guess V_{OH} is about 3V, meaning there is about 2V across the resistor. To get an I_B of 0.6mA, we need the resistor be less than or equal to 2V/0.6mA = 3.3kΩ. For a safety factor of 2 to 4, choose R to be 470Ω to 2kΩ. The 1N914 diode is required to remove the back-EMF generated during the PWM switching.

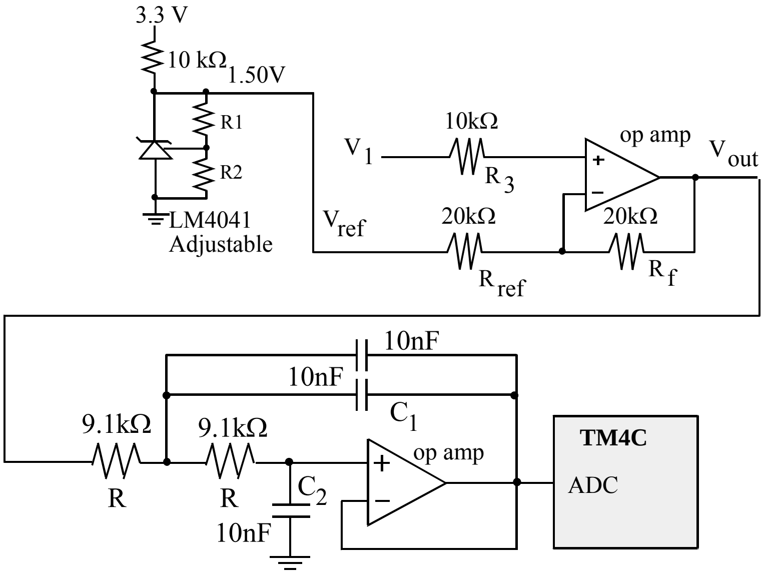




(5) Question 12. Consider a simplex synchronous serial interface from slave to master.



Notice, the data becomes available at 200ns, 200ns after the fall of the Clock. However, the data is required at 150ns, 100ns before the next rising edge. Data available does not overlap data required.



(15) Question 13. Interface this transducer.

$V_2 = 2V_1 - 1.5$

$V_2 = 2V_1 - V_{ref}$

No ground gain needed, already sum to +1

Choose a feedback resistor that is multiple of 1 and 2. Choose feedback resistor 20kΩ

V1 gain of 2 is 20kΩ/10kΩ

Vref gain of 1 is 20kΩ/20kΩ

first design step is to select the cutoff		
fc (Hz)	2500	fill this in
RA (kΩ)	10	same as initial R
C1A (µF)	0.009	is 141.4/(2•π•fc)
C2A (µF)	0.0045	is 70.7/(2•π•fc) or 0.5•C1A
second design step is to choose convenient Capacitor values		
fc (Hz)	2500	same as previous fc
RB (kΩ)	9.002	new value to match exact fc
C1B (µF)	0.01	fill this in
C2B (µF)	0.005	is 0.5•C1B
third design step is to choose a convenient resistor value		
fc (Hz)	2500.5	new cutoff based on these convenient values
RC (kΩ)	9.000	fill this value in
C1C (µF)	0.01	same as C1B
C2C (µF)	0.005	same as C2B