March 4, 2021, 3:30-4:45pm. This is an open book exam. You have 75 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

Question 1) This is book code for Timer2 periodic interrupts in section 6.2, which is similar to **ValvanoWareTM4C123v5\inc\Timer2A.c**

```
void (*PeriodicTask)(void);     // line 1
void Timer2A_Init(void(*task)(void), uint32_t period){ // line 2
  SYSCTL_RCGCTIMER_R |= 0x04;   // line 3
  PeriodicTask = task;          // line 4
  TIMER2_CTL_R = 0x00000000;    // line 5
  TIMER2_CFG_R = 0x00000000;    // line 6
  TIMER2_TAMR_R = 0x00000002;   // line 7
  TIMER2_TAILR_R = period-1;    // line 8
  TIMER2_TAPR_R = 0;            // line 9
  TIMER2_ICR_R = 0x00000001;    // line 10
  TIMER2_IMR_R = 0x00000001;    // line 11
  NVIC_PRI5_R = (NVIC_PRI5_R&0x00FFFFFF)|0x80000000; // line 12
  NVIC_EN0_R = 1<<23;           // line 13
  TIMER2_CTL_R = 0x00000001;    // line 14
  EnableInterrupts();           // line 15
}
void Timer2A_Handler(void){
  TIMER2_ICR_R = 0x00000001;    // line 16
  (*PeriodicTask)();            // line 17
}
```

**(5)** - Assume the PLL is not active, so the bus frequency is 16MHz. what will be the interrupt period in msec, if the initialization is called with a period parameter of xxx?
**(5)** - what is PeriodicTask?
**(5)** - change priority to highest level, meaning it is most important
**(5)** - what is the effect of executing line 16?

Question 2) This is book code for falling edge triggered interrupts in section 5.5. You will find a Port F version in **ValvanoWareTM4C123v5\inc\EdgeInterruptPortF.c**

```
volatile uint32_t FallingEdges = 0; // line 1
void EdgeCounter_Init(void){
  SYSCTL_RCGCGPIO_R |= 0x04;    // line 2
  FallingEdges = 0;             // line 3
  GPIO_PORTC_DIR_R &= ~0x10;    // line 4
  GPIO_PORTC_DEN_R |= 0x10;     // line 5
  GPIO_PORTC_IS_R &= ~0x10;     // line 6
  GPIO_PORTC_IBE_R &= ~0x10;    // line 7
  GPIO_PORTC_IEV_R &= ~0x10;    // line 8
  GPIO_PORTC_ICR_R = 0x10;      // line 9
  GPIO_PORTC_IM_R |= 0x10;      // line 10
  NVIC_PRI0_R = (NVIC_PRI0_R&0xFF00FFFF)|0x00A00000; // line 11
  NVIC_EN0_R = 4;               // line 12
  EnableInterrupts();           // line 13
}
void GPIOPortC_Handler(void){
  GPIO_PORTC_ICR_R = 0x10;      // line 14
  FallingEdges = FallingEdges + 1; // line 15
}
```

(5) - why does line 1 have a volatile?
(5) - change so it interrupts on both edges, rising and falling
(5) - is line 12 friendly?
(5) why is line 13 poor style, what would be better style?

Question 3) **(5)** Consider the IoT system from Lab 4 that communicates between the TM4C123 and the Blynk app on the phone. When I test my Lab 4, both my phone (running Blynk app) and the ESP8266 connect to the same wifi hot spot. When data are passed between my phone and the IoT device, do they pass through the cloud or can the data directly pass between them through the same hotspot?

Question 4) **(5)** Consider the IoT system from Lab 4 that communicates between the TM4C123 and the Blynk app on the phone. It uses a 64-character authorization code to establish a connection between the Blynk app and the IoT device. Does the fact that this is a 64-character code make the connection secure?
No, since Blynk does not disclose how the communication is passed, we cannot be sure.
No, 64-character code is easy to crack
Yes, since Blynk does not disclose how the communication is passed, no one will be able to crack it.
Yes, since Blynk uses TCP, it is secure
Yes, 64-character code is hard to crack

Question 5) **(5)** What is the complex impedance of a capacitor? Let j be the square root of -1, and f be the frequency.

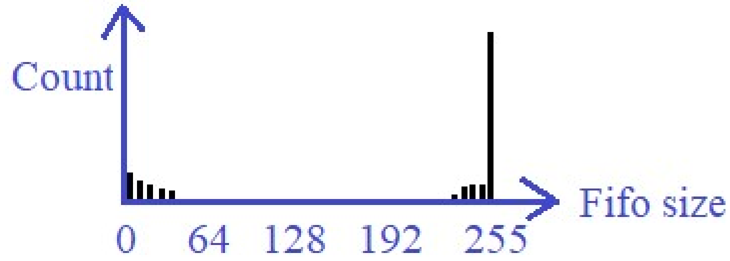
Question 6) **(5)** List all the ways you could **measure power line noise**

- Multimeter in AC voltage mode
- Oscilloscope in AC mode

Distractors

- Multimeter in DC voltage mode
- Multimeter in DC current mode
- Logic analyzer

Question 7) (5). Consider a producer consumer system that uses a 256-element FIFO to pass data from the producer to the consumer. The FIFO can store up to 255 elements. The consumer is software, processing data, getting the data from the FIFO. The producer is an interrupt-driven **input** device, which reads data from an I/O device and puts the data into the FIFO. We have added debugging instruments to the system in an effort to improve performance. After every time we put data into the FIFO, we measure the number of elements in the FIFO before attempting to put, and we create a probability mass function (pmf) of the behavior. If the FIFO size less than 255 when we call put, the FIFO saves the data. If the FIFO size is 255 when we call put, the FIFO is full and data are lost. Consider the following **pmf** of actual measurements over a long period of time.



How would you classify the system?

- I/O bound
- CPU bound

What is the best way to improve the system performance? Select the answer that guarantees a major improvement of performance.

- we could improve system bandwidth by increasing I/O speed
- we could eliminate lost data by increasing FIFO size
- we could improve system bandwidth by running a faster software algorithm that processes the data

**Question 8) (5)** Consider the interaction between this ISR and this main program. You may assume both the ISR and main are active and running

```
void SysTickHander(void){
  GPIO_PORTF_DATA_R ^= 0x02;
  GPIO_PORTF_DATA_R ^= 0x02;
// other stuff
  GPIO_PORTF_DATA_R ^= 0x02;
}
```

```
void main(void){
  Init();
  EnableInterrupts();
  while(1){
    GPIO_PORTF_DATA_R ^= 0x04;
// other stuff
  }
}
```

Do these read modify write accesses to port F data create a **critical section**?

Yes because the data register is shared and the toggle in the main program is nonatomic

Yes because the ISR does not disable interrupts during execution

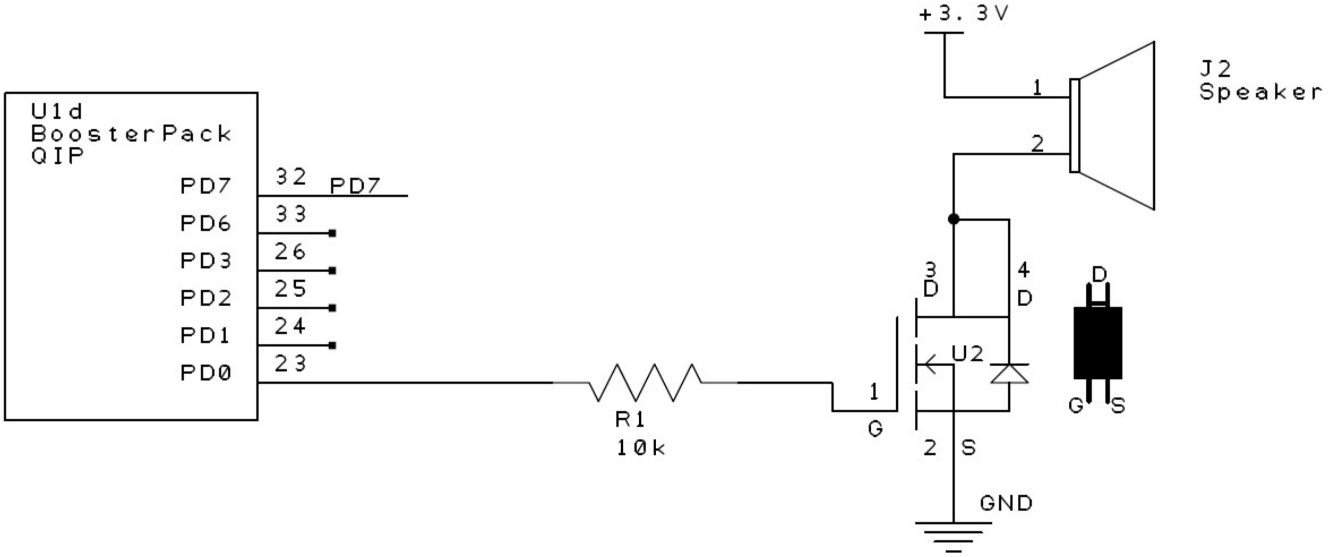
Yes because the read modify write sequence uses different bits

No because the read modify write sequence uses different bits

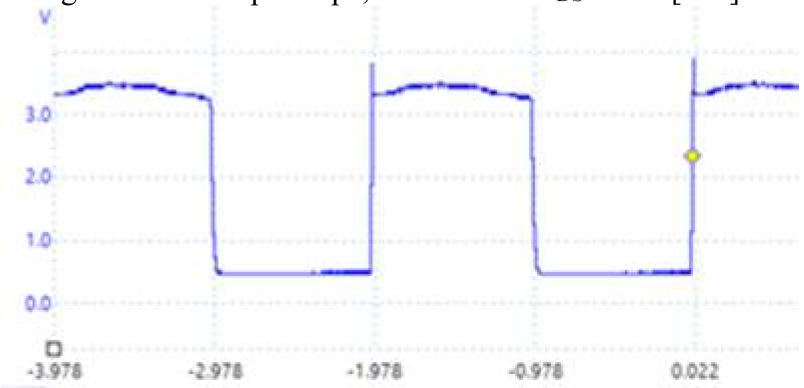
No because the ISR does not disable interrupts during execution

No because the toggle in the ISR is atomic

**Question 9) (5)** Consider this interface between the microcontroller pin PD0 and an 8-ohm speaker:



Using an oscilloscope scope, we measure $V_{DS}$ to be [xxx] V when the GPIO pin PD0 is high



,

How much power (W) is applied to the speaker when the GPIO pin is high?
V= (3.3- $V_{DS}$ ) in volts
I = (3.3- $V_{DS}$ )/8 in amps
P = (3.3- $V_{DS}$ ) * (3.3- $V_{DS}$ )/8 in watts

**Question 10) (5)** Consider a signed 16-bit binary fixed-point number system. The resolution is 2^n, where n is (-5 to -8). What is the **precision** of this number system in alternatives?
Answer = 65536

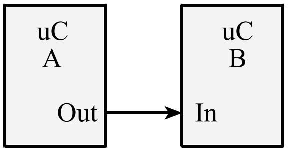
**(5) Question 11.** These are the parameters of the GPIO pins on ***microcontroller A***:

$I_{OLA}$, $I_{OHA}$, $I_{ILA}$, $I_{IHA}$, $V_{OLA}$, $V_{OHA}$, $V_{ILA}$, $V_{IHA}$

These are different parameters of the GPIO pins on ***microcontroller B***:

$I_{OLB}$, $I_{OHB}$, $I_{ILB}$, $I_{IHB}$, $V_{OLB}$, $V_{OHB}$, $V_{ILB}$, $V_{IHB}$

You wish to directly connect a GPIO output from microcontroller A to a GPIO input on microcontroller B.



List all the inequalities that must be true for the above connection to operate properly. Select all that apply

**$I_{OHA}$ greater than or equal to $I_{IHB}$**
**$I_{OLA}$ greater than or equal to $I_{ILB}$**
**$V_{OHA}$ greater than or equal to $V_{IHB}$**
**$V_{OLA}$ less than or equal to $V_{ILB}$**
$I_{OHA}$ less than or equal to $I_{IHB}$
$I_{OLA}$ less than or equal to $I_{ILB}$
$V_{OHA}$ less than or equal to $V_{IHB}$
$V_{OLA}$ greater than or equal to $V_{ILB}$

**Question 12. (10) Match the definition with the term.**

Give the definition and one example of a **firm real-time system**.

**arm** Activate so that interrupts are requested.
**atomic** Software execution that cannot be divided or interrupted.
**binary semaphore** A shared global variable that can have two values.
**cohesion** The degree to which all parts of the module are related to each other to satisfy a common objective.
**coupling T**he influence one module's behavior has on another module.
**friendly** Software modifies just the bits that need to be modified, leaving the other bits unchanged, making to easier to combine modules
**intrusive** The debugging itself significantly affects the program being tested.
**kibibyte** 1024 bytes.
**open drain** A digital logic output that has two states low and HiZ
**socket** An application endpoint for communication that encapsulates IP address, the transport protocol and the port

**Question 13.** (5) Match

**firm real-time** A system that expects all critical tasks to complete on time. Once a deadline as passed, there is no value to completing the task. However, the consequence of missed deadlines is real but the overall system operates with reduced quality.

**hard real-time** A system that can guarantee that a process will always complete a critical task within a certain specified range of time.

**soft real-time** A system that implements best effort to execute critical tasks on time, typically using a priority scheduler. Once a deadline as passed, the value of completing the task diminishes over time.

Parameters for the TM4C123 microcontroller (with 8mA mode selected)

$I_{OL}$ = 8mA, $I_{OH}$ = 8mA, $I_{IL}$ = 2µA, $I_{IH}$ = 2µA,

$V_{OL}$ = 0.4V, $V_{OH}$ = 2.4V, $V_{IL}$ = 1.3V, $V_{IH}$ = 2.0 V

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Name |
|---|---|---|---|---|---|---|---|---|
| DATA | DATA | DATA | DATA | DATA | DATA | DATA | DATA | GPIO_PORTB_DATA_R |
| DIR | DIR | DIR | DIR | DIR | DIR | DIR | DIR | GPIO_PORTB_DIR_R |
| IS | IS | IS | IS | IS | IS | IS | IS | GPIO_PORTB_IS_R |
| IBE | IBE | IBE | IBE | IBE | IBE | IBE | IBE | GPIO_PORTB_IBE_R |
| IEV | IEV | IEV | IEV | IEV | IEV | IEV | IEV | GPIO_PORTB_IEV_R |
| IME | IME | IME | IME | IME | IME | IME | IME | GPIO_PORTB_IM_R |
| RIS | RIS | RIS | RIS | RIS | RIS | RIS | RIS | GPIO_PORTB_RIS_R |
| MIS | MIS | MIS | MIS | MIS | MIS | MIS | MIS | GPIO_PORTB_MIS_R |
| ICR | ICR | ICR | ICR | ICR | ICR | ICR | ICR | GPIO_PORTB_ICR_R |
| SEL | SEL | SEL | SEL | SEL | SEL | SEL | SEL | GPIO_PORTB_AFSEL_R |
| DRV2 | DRV2 | DRV2 | DRV2 | DRV2 | DRV2 | DRV2 | DRV2 | GPIO_PORTB_DR2R_R |
| DRV4 | DRV4 | DRV4 | DRV4 | DRV4 | DRV4 | DRV4 | DRV4 | GPIO_PORTB_DR4R_R |
| DRV8 | DRV8 | DRV8 | DRV8 | DRV8 | DRV8 | DRV8 | DRV8 | GPIO_PORTB_DR8R_R |
| ODE | ODE | ODE | ODE | ODE | ODE | ODE | ODE | GPIO_PORTB_ODR_R |
| PUE | PUE | PUE | PUE | PUE | PUE | PUE | PUE | GPIO_PORTB_PUR_R |
| PDE | PDE | PDE | PDE | PDE | PDE | PDE | PDE | GPIO_PORTB_PDR_R |
| SLR | SLR | SLR | SLR | SLR | SLR | SLR | SLR | GPIO_PORTB_SLR_R |
| DEN | DEN | DEN | DEN | DEN | DEN | DEN | DEN | GPIO_PORTB_DEN_R |
| CR | CR | CR | CR | CR | CR | CR | CR | GPIO_PORTB_CR_R |
| AMSEL | AMSEL | AMSEL | AMSEL | AMSEL | AMSEL | AMSEL | AMSEL | GPIO_PORTB_AMSEL_R |

IS=0 means edge, IS=1 means level
IBE=1 means both, IBE=0 means one
If IBE=0, IEV=1 means rising, IEV=0 means falling

| Address | 31 – 29 | 23 – 21 | 15 – 13 | 7 – 5 | Name |
|---|---|---|---|---|---|
| 0xE000E400 | GPIO Port D | GPIO Port C | GPIO Port B | GPIO Port A | NVIC_PRI0_R |
| 0xE000E404 | SSI0, Rx Tx | UART1, Rx Tx | UART0, Rx Tx | GPIO Port E | NVIC_PRI1_R |
| 0xE000E408 | PWM Gen 1 | PWM Gen 0 | PWM Fault | I2C0 | NVIC_PRI2_R |
| 0xE000E40C | ADC Seq 1 | ADC Seq 0 | Quad Encoder | PWM Gen 2 | NVIC_PRI3_R |
| 0xE000E410 | Timer 0A | Watchdog | ADC Seq 3 | ADC Seq 2 | NVIC_PRI4_R |
| 0xE000E414 | Timer 2A | Timer 1B | Timer 1A | Timer 0B | NVIC_PRI5_R |
| 0xE000E418 | Comp 2 | Comp 1 | Comp 0 | Timer 2B | NVIC_PRI6_R |
| 0xE000E41C | GPIO Port G | GPIO Port F | Flash Control | System Control | NVIC_PRI7_R |
| 0xE000ED20 | SysTick | PendSV | -- | Debug | NVIC_SYS_PRI3_R |

| Address | | 30 | | 19 | | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Name |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xE000E100 | | F | | Timer0A | | UART1 | UART0 | E | D | C | B | A | NVIC_EN0_R |
| 0xE000E104 | | | | | | | | | | | UART2 | | NVIC_EN1_R |

| Address | 31-24 | 23-17 | 16 | 15-3 | 2 | 1 | 0 | Name |
|---|---|---|---|---|---|---|---|---|
| $E000E010 | 0 | 0 | COUNT | 0 | CLK_SRC | INTEN | ENABLE | NVIC_ST_CTRL_R |
| $E000E014 | 0 | 24-bit RELOAD value | | | | | | NVIC_ST_RELOAD_R |
| $E000E018 | 0 | 24-bit CURRENT value of SysTick counter | | | | | | NVIC_ST_CURRENT_R |