Jonathan W. Valvano First:_____ Last:_____

March 1, 2018, 3:30pm-4:45pm. This is a closed book exam, with one 8.5 by 11-inch crib sheet. You have 75 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(10) Question 1. You are asked to consult on a project to identify possible problems. You see the following write accesses to a shared global structure. These three software segments exist in different modules, called from interrupt service routines with three different priorities. These are the only accesses to the shared global structure. Initially, all fields of **`Status`** are 0.

```
typedef struct {
  int32_t a,b,x,y,z,flag;
} status_t;
status_t Status;
```

```
// low priority ISR
  Status.x = 5;
  Status.flag = 1;
```

```
// middle priority ISR
  Status.flag = 1;
  Status.y = 6;
```

```
// high priority ISR
  Status.a = 7;
  Status.flag = 1;
  Status.b = 8;
```

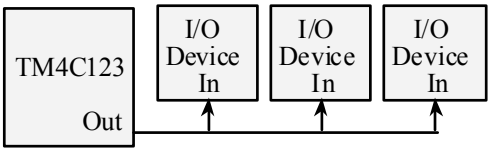
Do these accesses to **`Status`** create critical sections? Circle your choice and justify

Yes, they are *critical*

No, they are *not critical*

Justify your answer

(5) Question 2. You wish to connect a TM4C123 output pin (Out) to the input pins (In) of three identical I/O devices. The software selects 2-mA mode. The parameters of the I/O device are V_{IH} = 2.2V, I_{IH} = 0.1mA, V_{IL} = 0.7V, I_{IL} = 1mA. Will this interface operate properly? Circle your choice:



Yes, it will definitely operate properly, 100 % sure.

No, it will probably not work. It would work however, if 8-mA mode were selected.

No, it will definitely not operate properly, 100 % sure. This interface needs a driver

(10) Question 3. Consider an ideal capacitor and an ideal inductor. State the differential equations for each, relating voltage *V*, current *I*, inductance *L*, and capacitance, *C*.

Capacitor

Inductor

(10) Question 4. You wish to create a 32-bit binary fixed-point number system that can hold values from -1 to +1. Your system must include the value -1. Your system defines values up to +1, but does not include +1.

(3) Part a) What C data type should you use?

(4) Part b) What is the smallest (best) resolution that you could use?

(3) Part c) What integer would you store in memory if the value is -½? You can give the answer in decimal, hexadecimal, or exponential form.

(10) Question 5. The function **`CreateConnection`** calls the function **`sl_Htonl`** like this

```
Addr.sin_addr.s_addr = sl_Htonl(appData.DestinationIP);
```

The parameter is the 32-bit IP address of the server. The simple link function **`sl_Htonl`** is defined as

```
unsigned long sl_Htonl(unsigned long val){
  unsigned long i = 1;
  char *p = (char *)&i;
  if(p[0] == 1){
    p[0] = ((char*)&val)[3];
    p[1] = ((char*)&val)[2];
    p[2] = ((char*)&val)[1];
    p[3] = ((char*)&val)[0];
    return i;
  }
  else {
    return val;
  }
}
```

a) What does the function **`sl_Htonl`** do? In particular, define the input and output of the function.

b) Assume the if-condition is true. Assume the IP address is 192.168.1.0. What will be in the individual elements of the array **`p[]`** when **`sl_Htonl`** returns?

P[0]=	P[1]=	P[2]=	P[3]=

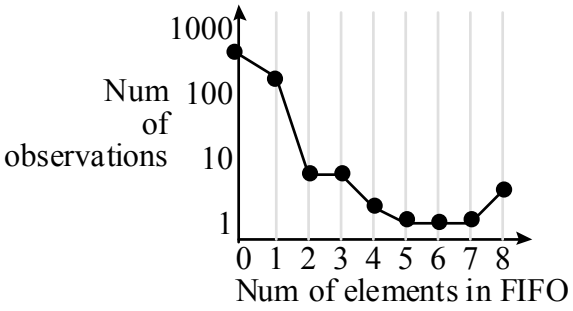
c) Explain the purpose of the if-else statement. When would the if-condition be false?

(5) Question 6. In Lab 4, how did we use the **`greet`** field within the **`Greeting`** class on the data-logging server? This is a piece of the Python code inside your server.

```
# [START greeting]
class Greeting(ndb.Model):
    """A main model for representing an individual Guestbook entry."""
    author = ndb.StringProperty(indexed=False)
    greet = ndb.StringProperty(indexed=False)
    date = ndb.DateTimeProperty(auto_now_add=True)
    city = ndb.StringProperty(indexed=False)
    ipaddr = ndb.StringProperty(indexed=False)
# [END greeting]
```

(5) Question 7. Does the servicing of an interrupt push the I-bit on the stack? If yes, explain why the I-bit needs to be saved. If no, explain why the I-bit should not be saved.

(10) Question 8. Consider an output device that uses a FIFO to pass data from main program (which puts) to the ISR (which gets). The FIFO can hold up to 8 elements. The size of the FIFO was measured periodically resulting in the following histogram.



(5) Part a) Is this I/O bound or CPU bound (circle your answer)

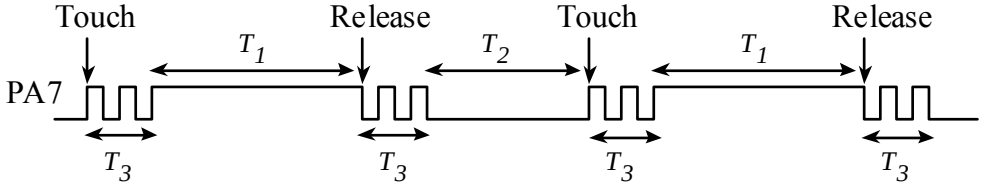
I/O bound

CPU bound

(5) Part b) Should you increase the size of the FIFO? If yes, explain why the FIFO size should be increased. If no, explain why this FIFO size is ok.

(10) Question 9. Interface an 8-Ω speaker to the microcontroller. A digital output squarewave on PB0 creates sound on the speaker. Assume PB0 is an output with 8-mA selected. If PB0 is high, a voltage between 6 and 8 volts should be applied across the speaker. Because of the 8-Ω impedance, the speaker requires up to 1 A to operate. If PB0 is low, no current should flow through the speaker. You may use +3.7V, +7.4V, +11.1V, or 14.4V power (choose the correct one to use). Decide to use no transistor ($0.00), a 2N2222 ($0.44), or a TIP120 ($0.72). Decide to use no diode ($0.00), or a 1N914 ($0.04). Select the least expensive circuit that will operate the speaker. Show your work including resistance values. No software needed, just the hardware circuit. *See reference sheet at the end of the quiz.*

(25) Question 10. A positive logic switch is interfaced to PA7. You may assume Port A is already initialized so PA7 is an input. The time T_1 that the input is high is a minimum of 20 ms. There is no maximum T_1. The time T_2 that the input is low is also a minimum of 20 ms. There is no maximum T_2. There is significant bounce on the switch. There may be 1, 3, or 5 edges each time the switch is touched or released. The bounce time, T_3, has a maximum of 5 ms. The minimum T_3 is 0 ms. This means the maximum touch rate is 25 times/sec. There is no minimum touch rate.



The goal of the problem is to execute a user function once per touch and execute another user function once per release. Rather than using edge triggered interrupts like class, you must use SysTick interrupts (and no other timers or interrupts). Assume bus clock is 16 MHz.

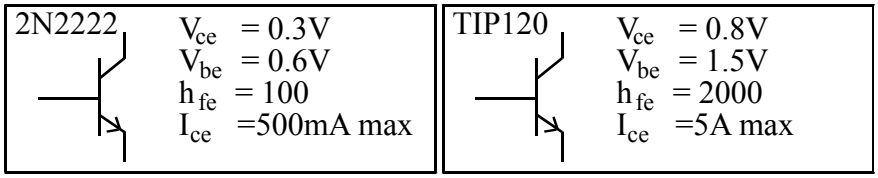
(12) Part a) Write the initialization function that configures the SysTick interrupts. Define any global variables and initialize them as needed. The prototype is

```
void Switch_Init(void(*T)(void), void(*R)(void));
```

where ***T** is a pointer to the touch function and ***R** is a pointer to the release function. Be careful to clearly specify the SysTick interrupt period. Set the interrupt priority to the most important level. Arm and enable interrupts.

(13) Part b) Write the SysTick interrupt service routine

Parameters for the TM4C123 microcontroller, with 2/4/8-mA mode selected
I_{OL} = 2/4/8mA, I_{OH} = 2/4/8mA, I_{IL} = 2μA, I_{IH} = 2μA,
V_{OL} = 0.4V, V_{OH} = 2.4V, V_{IL} = 1.3V, V_{IH} = 2.0 V



	31–3	2–0	Name
$4003.1000		GPTMCFG	TIMER1_CFG_R

	31–4	3	2	1-0	Name
$4003.1004		TAAMS	TACMR	TAMR	TIMER1_TAMR_R

	14	13	11-10	8	6	5	3-2	0	Name
$4003.100C	TBPWML	TBOTE	TBEVENT	TBEN	TAPWML	TAOTE	TAEVENT	TAEN	TIMER1_CTL_R

	31-11	10	9	8	7-4	2	1	0	Name
$4003.1018		CBEIM	CBMIM	TBTOIM		CAEIM	CAMIM	TATOIM	TIMER1_IMR_R

	31-11	10	9	8	7-4	2	1	0	Name
$4003.101C		CBERIS	CBMRIS	TBTORIS		CAERIS	CAMRIS	TATORIS	TIMER1_RIS_R

	31-11	10	9	8	7-4	2	1	0	Name
$4003.1020		CBECINT	CBMCINT	TBTOCINT		CAECINT	CAMCINT	TATOCINT	TIMER1_ICR_R

	31–16	15–0	Name
$4003.1028	TAILRH	TAILRL	TIMER1_TAILR_R

	31–8	7-0	Name
$4003.1038		TAPSR	TIMER1_TAPR_R

	31–8	7-0	Name
$4003.1040		TAPSMR	TIMER1_TAPMR_R

	31–16	15-0	Name
$4003.1048	TARH	TARL	TIMER1_TAR_R

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
0xE000E45C	Wide Timer 0B	Wide Timer 0A	Timer 5B	Timer 5A	NVIC_PRI23_R
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R

Address		30	23	21	19	5	4	3	2	1	0	Name
0xE000E100		F	Timer2A	Timer1A	Timer0A	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