Jonathan W. Valvano First:_________________ Last:_____________________

October 10, 2014, 10:00am-10:50am. This is a closed book exam, with one 8.5 by 11 inch crib sheet. You have 50 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(15) Question 1. This debugging instrument measures a histogram of count versus the number of FIFO elements stored. The index-based FIFO is index based, so **`(uint32_t)(TxPutI-TxGetI)`** returns the current number of elements stored in the FIFO, which can vary from 0 to 15. The macro **`TxProfile()`** is invoked just before every time the foreground calls **`TxFifo_Put`**. Thus, **`TxPDF[0]`** is the number of times the FIFO was empty at the time when put was called.

```
uint32_t TxPDF[16];
#define TxProfile() TxPDF[(uint32_t)(TxPutI-TxGetI)]++;
```

When the **`TxProfile`** macro is expanded it creates the following assembly code. According to the technical reference manual, the load and store instructions take two bus cycles and the addition and subtraction instructions take only one bus cycle. The software is running at 16 MHz (62.5ns).

```
0x0000075A 492E      LDR      r1,[pc,#184]  ; @0x00000814   [2]
0x0000075C 6809      LDR      r1,[r1,#0x00] ; TxPutI        [2]
0x0000075E 4A2E      LDR      r2,[pc,#184]  ; @0x00000818   [2]
0x00000760 6812      LDR      r2,[r2,#0x00] ; TxGetI        [2]
0x00000762 1A89      SUBS     r1,r1,r2      ; TxPutI-TxGetI [1]
0x00000764 4A2D      LDR      r2,[pc,#180]  ; @0x0000081C   [2]
0x00000766 EB020081  ADD      r0,r2,r1,LSL #2 ;             [1]
0x0000076A 6801      LDR      r1,[r0,#0x00]                 [2]
0x0000076C 1C49      ADDS     r1,r1,#1      ; increment PDF [1]
0x0000076E 6001      STR      r1,[r0,#0x00]                 [2]
```

Part a) What does **`TxPDF[1]`** represent?

Part b) There are on average 100,000 elements per second (ISR called every 10µs) being passed from the foreground (main program) to the background (ISR). Is this debugging instrument *nonintrusive*, *minimally intrusive*, or *highly intrusive*? **Justify your answer**.

Part c) Make a rough sketch of the data in **`TxPDF`** that would occur if the system were to be CPU bound.

Bonus) What would it mean if the **`ADDS`** instruction were to set the carry bit?

(5) Question 2. Consider the same debugging instrument from Question 1 in other context, where

```
   TxPDF[(uint32_t)(TxPutI-TxGetI)]++;
```

is invoked from multiple ISRs with different priorities. This code is not reentrant. Specify the critical section. Do not consider the interaction with `Fifo_Put` and `Fifo_Get`. Only consider this code as it interacts with threads that reenter this code. In particular, between which assembly instructions is the critical section? Do not circle instructions, rather place arrows between instructions that are critical.

```
0x0000075A 492E      LDR      r1,[pc,#184]  ; @0x00000814   [2]
0x0000075C 6809      LDR      r1,[r1,#0x00] ; TxPutI        [2]
0x0000075E 4A2E      LDR      r2,[pc,#184]  ; @0x00000818   [2]
0x00000760 6812      LDR      r2,[r2,#0x00] ; TxGetI        [2]
0x00000762 1A89      SUBS     r1,r1,r2      ; TxPutI-TxGetI [1]
0x00000764 4A2D      LDR      r2,[pc,#180]  ; @0x0000081C   [2]
0x00000766 EB020081  ADD      r0,r2,r1,LSL #2 ;             [1]
0x0000076A 6801      LDR      r1,[r0,#0x00]                 [2]
0x0000076C 1C49      ADDS     r1,r1,#1      ; increment PDF [1]
0x0000076E 6001      STR      r1,[r0,#0x00]                 [2]
```

(10) Question 3. These are the parameters of the GPIO pins on a 9S12DP512 microcontroller:

I_{OL} = 10mA, I_{OH} = 10mA, I_{IL} = 1μA, I_{IH} = 1μA,
V_{OL} = 0.8V, V_{OH} = 4.2V, V_{IL} = 1.75V, V_{IH} = 3.25 V

Part a) Can you interface a 9S12DP512 GPIO output to a TM4C12xx GPIO input? If yes, prove it. If no, show at least one parameter not satisfied.

Part b) Can you interface a TM4C12xx GPIO output to a 9S12DP512 GPIO input? If yes, prove it. If no, show at least one parameter not satisfied.

(25) Question 4. Make PB1 and PB2 inputs. You are given two tasks: the subroutine `Task1()` should be executed on the rising edge of PB1. It takes 10 to 100 μs to execute `Task1`. The subroutine `Task2()` should be executed on the rising edge of PB2. It takes 10 to 20 μs to execute `Task2`. Both tasks are equally important but you should minimize the sum of the latencies for both tasks. You may assume there are other interrupts with priorities 1, 2 and 3. Rising edges can occur at the same time, or close to each other. However, the time from rising edge to rising edge on that same signal is large compared to 100 μs.

Part a) Show the ritual to initialize this system. Fill in the boxes with an operator and a value to make the ritual friendly. The first one and last have been completed to show you what it should look like.

```
void PortB_Init(void){ unsigned long volatile delay;
  SYSCTL_RCGCGPIO_R      |= 0x02           ;
  delay = SYSCTL_RCGCGPIO_R;
  GPIO_PORTB_DIR_R                         ;
  GPIO_PORTB_DEN_R                         ;
  GPIO_PORTB_IS_R                          ;
  GPIO_PORTB_IBE_R                         ;
  GPIO_PORTB_IEV_R                         ;
  GPIO_PORTB_IM_R                          ;
  NVIC_PRI0_R = (NVIC_PRI0_R&0xFFFF00FF)|                    ;
  NVIC_EN0_R             = 0x00000002      ;
  EnableInterrupts();
}
```

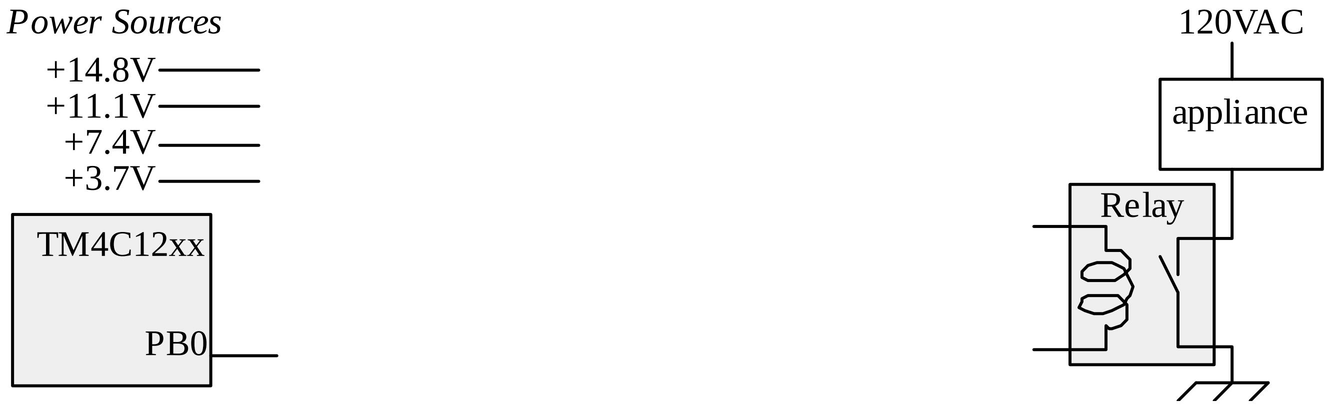
Part b) Show the interrupt service routine (Port B handler). No backward jumps are allowed.

```
void GPIOPortB_Handler(void){
```

(10) Question 5. Consider an ideal capacitor, with capacitance *C*. Let *V* be the voltage across the capacitor, and let *I* be the current through the capacitor. Give an equation that relates *C*, *I*, and *V*.

(10) Question 6. PWM is used to deliver variable power to a DC motor. The time constant of the motor is 10ms, so the PWM period is selected to be 1ms. The maximum speed of the motor is 100 rps. The bus clock is 16 MHz. What is the equivalent number of bits of precision for this actuator? In other words, how many choices of delivered power can this interface generate? Show your work.

(25) Question 7. Interface an electromagnetic relay to the microcontroller. A digital output on PB0 controls the relay. If PB0 is high, a voltage should applied across the relay coil (any voltage 6V±1.2V), and the 120VAC switch will activate causing the appliance to turn on. If PB0 is low, no current should flow through the relay coil, and the 120VAC switch will deactivate causing the appliance to turn off. The maximum relay coil current is 50mA. Decide whether to use the PN2222 ($0.21), 2N2222 ($0.44), or the TIP120 ($0.72). Select the least expensive transistor that will operate the relay. You may use +3.7V, +7.4V, +11.1V, and/or +14.8V power. Show your work including resistance values. No software needed, just the hardware circuit.



Parameter	PN2222 (I_C=150mA) PN2907 (I_C=150mA)	2N2222 (I_C=500mA) 2N2907 (I_C=500mA)	TIP120 (I_C=3A) TIP125 (I_C=3A)
h_{fe}	100	40	1000
V_{BEsat}	0.6	2	2.5 V
V_{CE} at saturation	0.3	1	2 V

Parameters for the TM4C12xx microcontroller (no 12mA mode will be used)

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

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
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R

Address	31	30	29-7	6	5	4	3	2	1	0	Name
0xE000E100	G	F	…	UART1	UART0	E	D	C	B	A	NVIC_EN0_R
0xE000E104			…						UART2	H	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

Address	31-29	28-24	23-21	20-8	7-5	4-0	Name
$E000ED20	TICK	0	PENDSV	0	DEBUG	0	NVIC_SYS_PRI3_R

	31–3	2–0	Name
$4003.0000		**GPTMCFG**	TIMER0_CFG_R

	31–4	3	2	1-0	
$4003.0004		**TAAMS**	**TACMR**	**TAMR**	TIMER0_TAMR_R

	14	13	11-10	8	6	5	3-2	0	
$4003.000C	TBPWML	TBOTE	TBEVENT	TBEN	TAPWML	TAOTE	**TAEVENT**	**TAEN**	TIMER0_CTL_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0018		CBEIM	CBMIM	TBTOIM		**CAEIM**	CAMIM	TATOIM	TIMER0_IMR_R

	31-11	10	9	8	7-4	2	1	0	
$4003.001C		CBERIS	CBMRIS	TBTORIS		**CAERIS**	CAMRIS	TATORIS	TIMER0_RIS_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0020		CBEMIS	CBMMIS	TBTOMIS		CAEMIS	CAMMIS	TATOMIS	TIMER0_MIS_R

	31-11	10	9	8	7-4	2	1	0	
$4003.0020		CBECINT	CBMCINT	TBTOCINT		**CAECINT**	CAMCINT	TATOCINT	TIMER0_ICR_R

	31–16	15–0	
$4003.0028	TAILRH	**TAILRL**	TIMER0_TAILR_R

	31–16	15–0	
$4003.0030	TAMRH	TAMRL	_TAMATCHR_R

	31–8	7-0	
$4003.0038		TAPSR	TIMER0_TAPR_R

	31–8	7-0	
$4003.0040		TAPSMR	TIMER0_TAPMR_R

	31–16	15-0	
$4003.0048	TARH	**TARL**	TIMER0_TAR_R