Department of Electrical and Computing Engineering

UNIVERSITY OF CONNECTICUT

ECE 3411 Microprocessor Application Lab: Fall 2015

Quiz IV

There is 1 questions in this quiz. There are 15 pages in this quiz booklet. Answer each question according to the instructions given.

You have **45 minutes** to answer the questions.

Some questions are harder than others and some questions earn more points than others—you may want to skim all questions before starting.

If you find a question ambiguous, be sure to write down any assumptions you make.
Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name in the space below. Write your initials at the bottom of each page.

THIS IS A CLOSED BOOK, CLOSED NOTES QUIZ.
PLEASE TURN YOUR NETWORK DEVICES OFF.

Any form of communication with other students is considered cheating and will merit an F as final grade in the course.

Do not write in the boxes below

1 (x/40)	2 (x/40)	3(A) (x/20)	3(B) (x/40)	3(C) (x/20)	3(D) (x/20)	Total (xx/180)

Name:

Student ID:

1. [40 points]: Assume a clock frequency of $f_{clk} = 20\text{MHz}$ and the following initialization:

```
DDRD   = 0x10;
OCR1A  = 39062;
OCR1B  = 13020;
TCCR1A = 0b00110011;
TCCR1B = 0b00011101;
```

Answer the following questions:

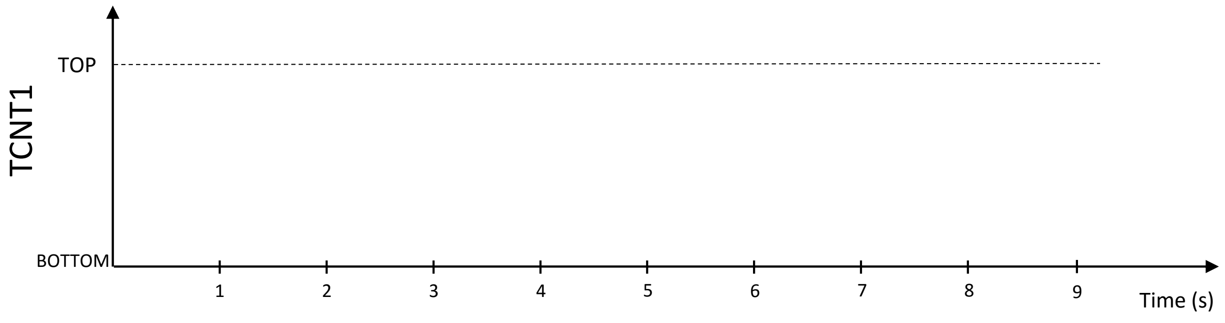
a. In which mode is `Timer1` running?

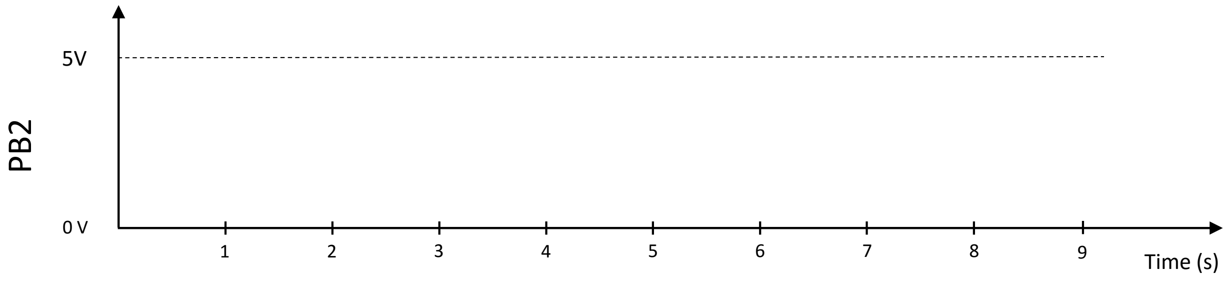
b. What is the numerical value of 'TOP' for `Timer1` in this mode?

c. How much time (in seconds) does it take for `Timer1` to complete one full cycle, i.e. going from BOTTOM $\rightarrow$ TOP $\rightarrow$ BOTTOM? Be as accurate as possible in your calculations.

Initials:

d. Starting from the moment of `Timer1`'s initialization, draw the waveforms of the `TCNT1` register value and the pin PB2 value w.r.t. time. Please draw the waveform strictly according to the timing scale shown on X-axis, otherwise no credit will be given.





e. In each full cycle of `Timer1`:

- For how much time (in seconds) is PB2 low? Be as accurate as possible in your calculations.

- For how much time (in seconds) is PB2 high? Be as accurate as possible in your calculations.

Initials:

2. [40 points]: Given that the clock frequency ($clk_{I/O}$) of ATmega328P is 16MHz, implement the finite state machine (FSM) shown in Figure 1. The state transitions are made whenever a button connected to INT0 pin (i.e. PD2) is pushed and a **Falling Edge** is detected at INT0. Each state produces an output signal at PB2, and the output specifications of the states are as follows:

State_A: PB2 stays at logic LOW level.

State_B: A non-inverting 1kHz PWM signal with 30% duty cycle is generated at PB2.

State_C: A non-inverting 2kHz PWM signal with 70% duty cycle is generated at PB2.

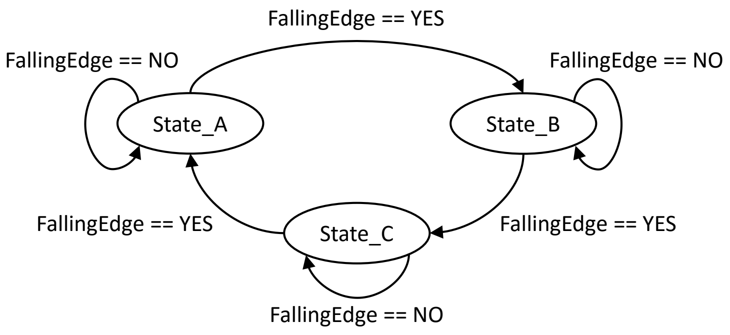


Figure 1: State Transition Diagram of the FSM

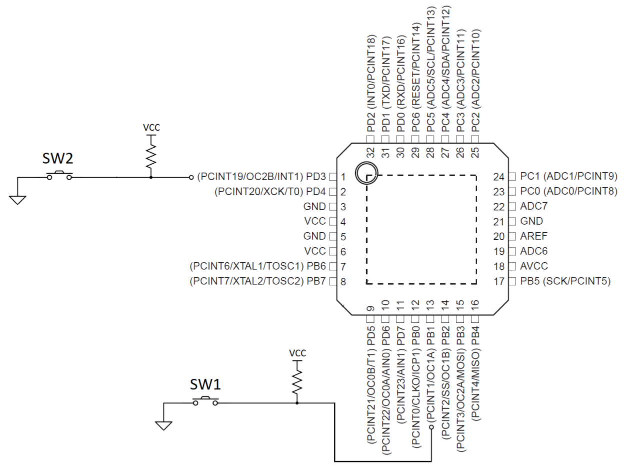


Figure 2: ATmega328P Hardware Configuration.

Assuming that the push button does not need debouncing, complete the following code segments.

Initials:

The following code snippet provides the necessary includes, declarations, definitions and a basic layout.

```
/*********** ECE3411 Quiz 5, Task 2 ************/
#define F_CPU 16000000UL
#include <avr/io.h>
#include <inttypes.h>
#include <avr/interrupt.h>

// For State Machine
#define State_A    1
#define State_B    2
#define State_C    3
volatile uint8_t System_State;

// For PWM
volatile uint16_t time_period;
volatile uint16_t duty_cycle;

// Define any other variables here



//-----------------------------------------------------------------------

/* Triggers at Falling Edge on PD2 */
ISR(INT0_vect)
{
    // Calls state transition function
    make_state_transition();
}
//-----------------------------------------------------------------------

// Timer 1 Compare Match A ISR (TCNT1 = OCR1A)
ISR (TIMER1_COMPA_vect)
{
    OCR1A = time_period;   // Update PWM time period
    OCR1B = duty_cycle;    // Update PWM duty cycle
}
//-----------------------------------------------------------------------

/* Main Function */
int main(void)
{
    initialize_all();     // Initialize everything
    sei();                // Enable Global Interrupts
    while(1);             // Nothing to do.

} /* End of main() */
```

Initials:

3. [100 points]: A colleague wants your help in executing a particular task for which you need to write a code such that a task() is executed as soon as the following events occur (See Figure 1).

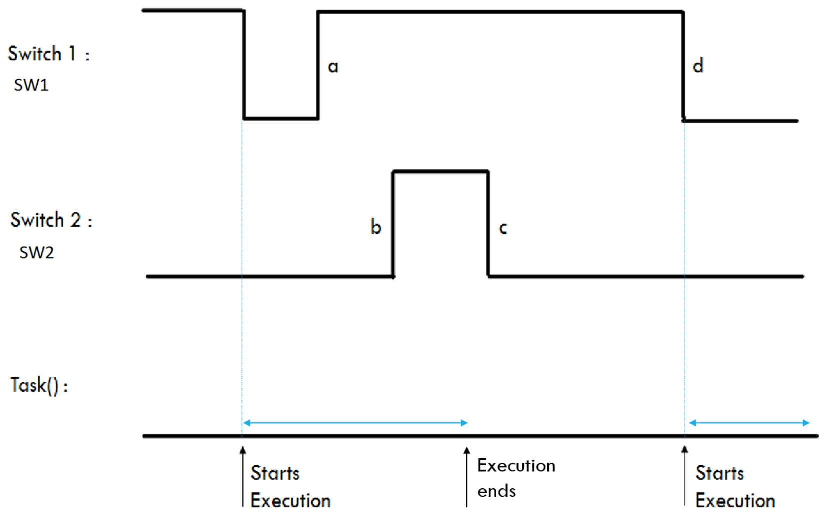


Figure 3: Timing Diagram.

a - There is a rising edge at SW1

b, c - SW2 toggles twice after event (a)

d - After events b and c occur there is a falling edge at SW1 and task() is not running at that very moment. Note that if task() is executing at that moment then the MCU needs to wait for event (a) to occur again.

The switches SW1 and SW2 are connected to PB1 and PD3 of ATmega328P respectively, as shown in the Figure 4. The clock frequency ($clk_{I/O}$) is 16MHz.

Implement this system by answering the short questions and filling in the gaps in the code layout given below. Notice that you are **not allowed** to use any software counter or `_delay_ms()`/`_delay_us()` routines.

Initials:

The following code layout needs to be used.

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <inttypes.h>
#include <avr/interrupt.h>
// Flag Variables
volatile uint8_t taskflag;

// Declare more variables as required in sub problem [B]

void initialize_all(void)
{
    //To be filled in sub problem [A]
}

ISR( //To be filled in sub problem [A]  )
{
   //To be filled in sub problem [B]
}


ISR( //To be filled in sub problem [A]  )
{
   //To be filled in sub problem [B]
}

/* Main Function */
int main(void)
{
    initialize_all();     // Initialize everything
    sei();                // Enable Global Interrupts
    taskflag = 0;

    while(1)
    {
        if(taskflag == 1)
        {
           // In sub problem [B] you will need to decide the order of
           execution of the statements 1) taskflag = 0; and 2) task();

        }
    }

} /* End of main() */
```

Initials:

A. Initialization: (20 points)
Complete the function `initialize_all(void)` as instructed below:

```
/* Initialization function */
void initialize_all(void)
{

    // Program only the necessary control register and ports
















} /* End of initialize_all() */
```

Give the names of the interrupt vector used for interpreting the input from the 2 connected switches.(see figure 2 provided at the end of the quiz)

(a) ISR(C)
(b) ISR(D)

C -

D -

Initials:

B. Interrupt ISR: (40 points)
Complete the function `ISR( C )` and `ISR( D )` and declare the necessary variables. **Do not** execute the task() in the ISR, instead set the taskflag value accordingly. [Hint : It would be helpful to use a FSM that tracks the event sequence.]

```
  // Declarations
```

```
ISR( C )
{

   // Code

}
```

Initials:

```
ISR( D )
{

    //Code

























}
```

Write the code for the while loop in the main function

```
while(1)
{
   if(taskflag == 1)
   {

      //Complete code here




   }
}
```

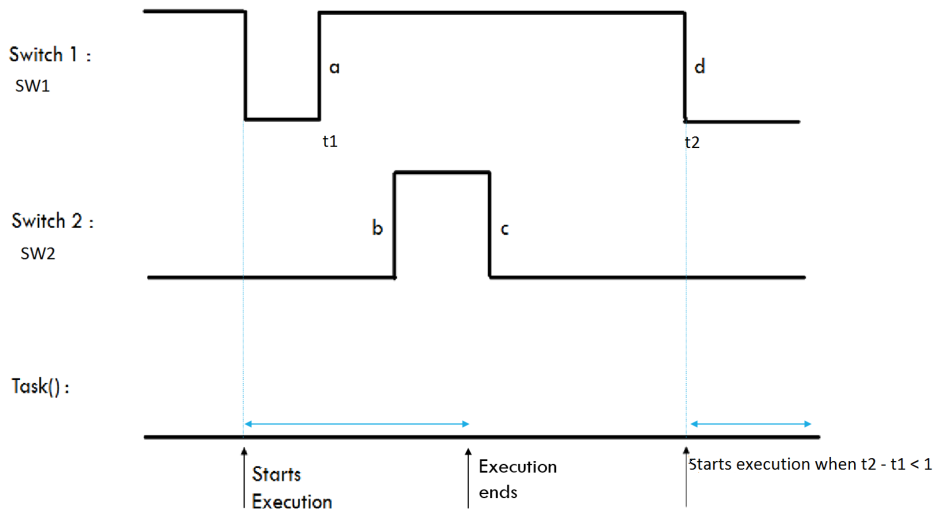
Initials:

C. Extend the system to implement an additional requirement(20 points)

Consider the timing diagram given below. Suppose event (a) occurs at time t1 and event (d) occurs at time t2.

You are asked to change the code such that when event (d) happens you also check whether the time $t2 - t1$ is less than 1 second. If this is not the case the task() will not be executed.

Describe in words what changes need to be included in the code. [Hint : Think about how you can measure the time between the events a and b, the extra declarations required etc.]



Initials:

D. (20 points) Suppose the switch SW2 is connected to PB7. Explain in words what changes you need to incorporate in the code?

Initials:

You may use the provided data sheets for your reference.

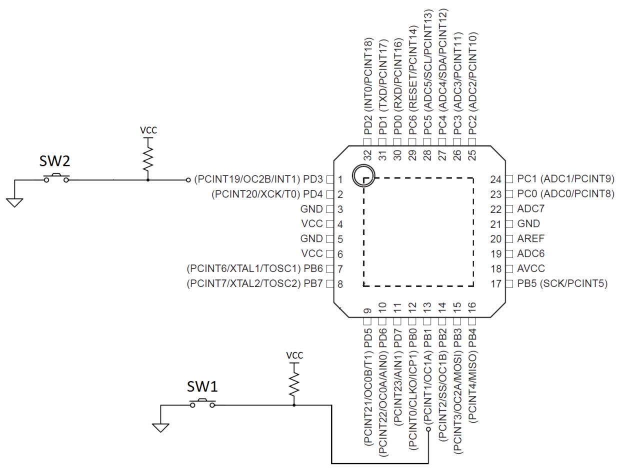
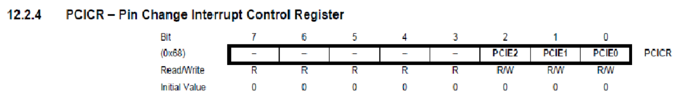
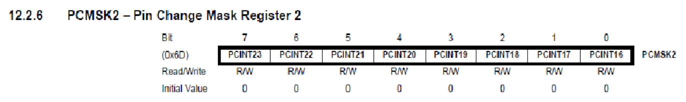


Figure 4: ATmega328P Hardware Configuration.



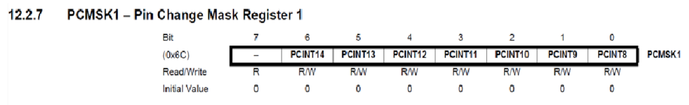
12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



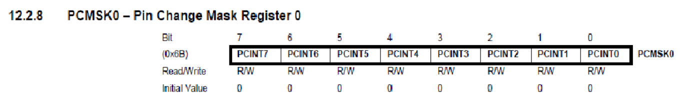
12.2.6 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



12.2.7 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

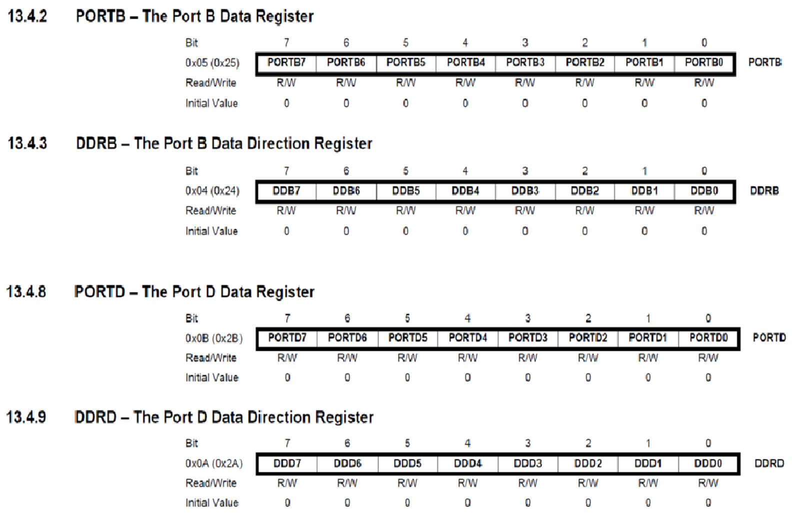


12.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Initials:

.



13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Initials:

End of Quiz

Please double check that you wrote your name on the front of the quiz.

Initials: