Department of Electrical and Computing Engineering

UNIVERSITY OF CONNECTICUT

ECE 3411 Microprocessor Application Lab: Fall 2015

Quiz V

There are 3 questions in this quiz. There are 10 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 (x/20)	Total (xx/100)

Name:

Student ID:

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

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

Answer the following questions:

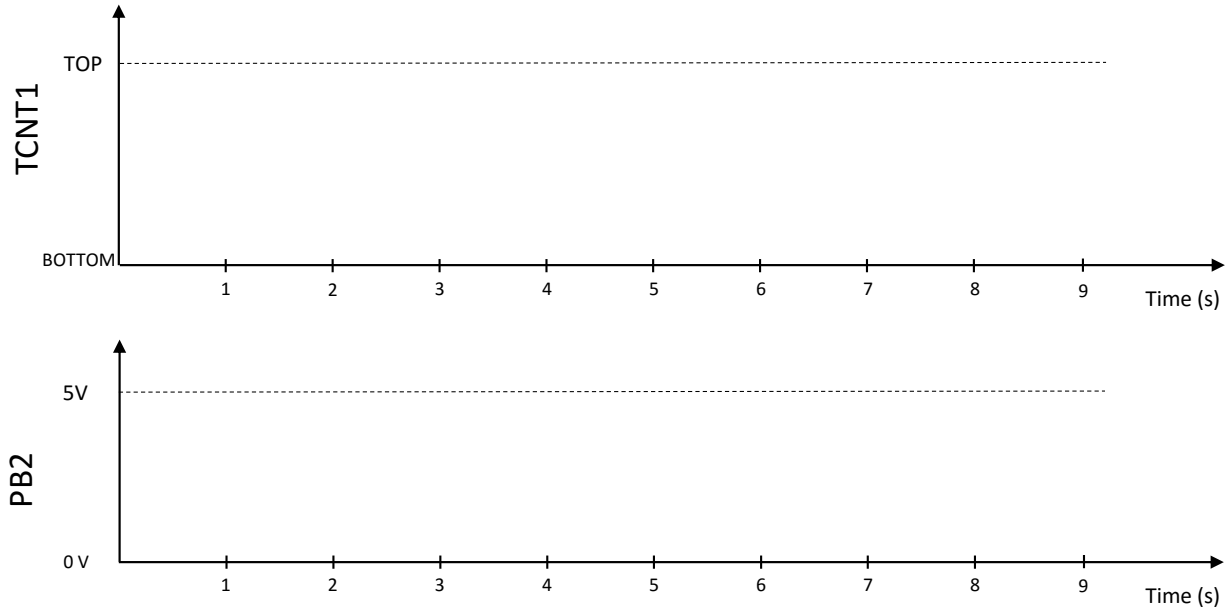
a. In which mode is `Timer1` running?

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.

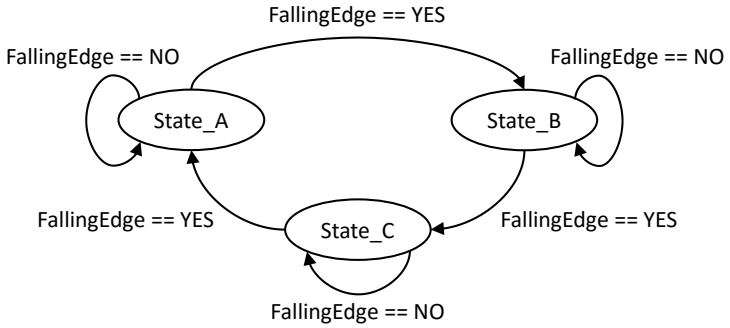


Figure 1: State Transition Diagram of the FSM

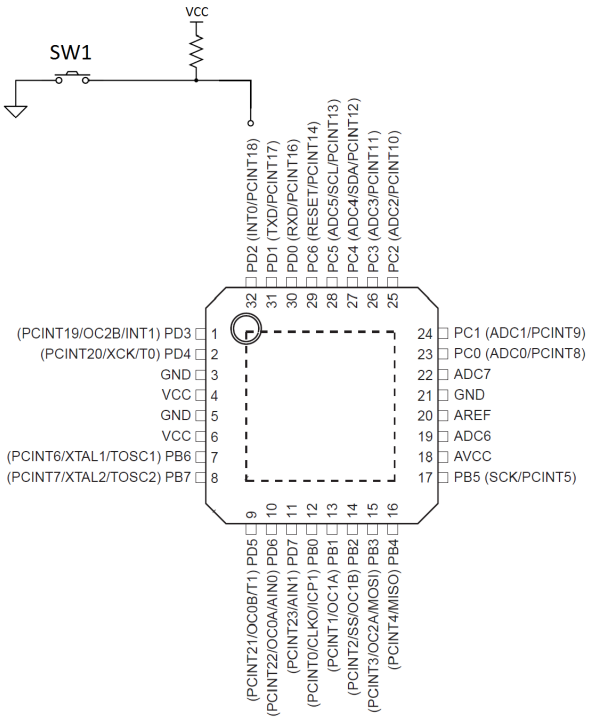


Figure 2: ATmega328P Hardware Configuration.

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:

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

```
/* Initialization function */
void initialize_all(void)
{
    // Initializing the state variable
    System_State = State_A;

    /* Configure PB2 here */



    /* Configure INT0 here */









    /* Configure Timer 1 here */

















    /* Any other initializations here if needed */









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

Initials:

B. State Transition Function Implementation: (20 points)
Write the function `make_state_transition()` to implement the FSM.

```
/* State transition function called by INT0 ISR */
void make_state_transition()
{

} /* end of make_state_transition() */
```

Initials:

3. [20 points]: Given that clock frequency ($clk_{I/O}$) of ATmega328P is 8MHz.
Assume the following about the code snippet given below:

- Each one of `instruction_1`, `instruction_2`, . . . , `instruction_52` takes 4 CPU cycles.
- Evaluating `while(1)` statement takes zero CPU cycles.
- Evaluating `if( !(ADCSRA & (1<<ADSC)) )` statement and executing its body take zero CPU cycles.

```
/************ ECE3411 Quiz 5, Task 3 *************/
#define F_CPU 8000000UL
#include <avr/io.h>

/* Main Function */
int main(void)
{
    /* Configuring ADC Control and Status Register A */
    ADCSRA = 0x86;

    while(1)
    {
        instruction_1;
        instruction_2;
        instruction_3;
        ...
        ...
        instruction_52;
        if( !(ADCSRA & (1<<ADSC)) )
        {
            ADCSRA |= (1<<ADSC);    // Start A to D Conversion
        }

    } /* End of while(1) Loop */

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

Answer the following questions about the code snippet.

Initials:

a. Given that it takes 13 ADC cycles, how much time (in microseconds) does it take to complete one ADC conversion?

b. What prevents the condition "`if( !(ADCSRA & (1<<ADSC)) )`" from being satisfied?

c. How much time (in microseconds) does it take to complete one iteration of "`while(1)`" loop?

d. What is the percentage of `while(1)` loop iterations for which the body of "`if( !(ADCSRA & (1<<ADSC)) )`" condition is executed?

Initials:

End of Quiz

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

Initials: