

# CPE 325: Embedded Systems Laboratory

## Laboratory Assignment #6

### Assignment

[100 pts]

- Write an **Assembly Program** that interfaces switches, SW1 and SW2, and LEDs, LED1 and LED2, as follows (**You should use interrupts for both switches**). [50 pts]
  - Initially, both the LEDs should be turned on.
  - When SW1 is pressed for the first time, LED2 should be turned off. The next time SW1 is pressed, LED2 is turned on. The sequence continues hence each press changes the state of LED2.
  - SW2 controls LED1 and LED2 as follows: when SW2 is pressed, LED1 blinks 4 times at 1 Hz, then toggle LED2.

**Hint:** Inside the ISR change the edge of P1IES to trigger the ISR as required.

- Consider the **C program** given below: [50 pts]

```
#include <msp430.h>
void main(void) {
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    P1DIR |= BIT0; // Set LED1 as output
    P1OUT = 0x00; // Clear LED1 status
    P4DIR |= BIT7; // Set LED2 as output
    P4OUT = 0x80; // Clear LED2 status

    unsigned int i = 0;
    while(1) {
        for(i=0; i<50000; i++);
        P1OUT ^= 0x01; // Toggle LED1 status
        P4OUT ^= 0x80; // Toggle LED2 status
    }
}
```

Update the code given above to meet the following requirements: Every time when SW2 is pressed, the clock frequency set to 8 MHz, and Every time when SW1 is pressed, the frequency of the clock is decreased by half. The clock frequency should not go below 1 MHz.

### Topics for Theory:

- Interrupts and interrupt vectors - how do they work, what can you use them for?
- Clock module in MSP430 - what is it, and how can you use it to change clock frequencies?

## Deliverables

1. Lab report:
  - a. Theory,
  - b. Screenshots of final outputs,
  - c. Flowchart for Q.2 and
  - d. Frequency calculation
2. Source files (.asm files and C files)

## Note

1. Do not implement a loop in ISR.
2. Implement debouncing properly.