CPE/EE 323 Introduction to Embedded Computer Systems
Homework V

1 (15)	2 (25)	3 (25)	4 (20)	5 (15)	Total

Problem #1 (15 points) Exception Processing

Answer the following questions.

A. (2 points) Illustrate how the top of the stack should look like at the moment just before RETI is executed.

B. (5 points) The MSP430F5529 receives interrupt requests from the watchdog timer in the interval mode (WDT) and parallel ports P1.7 and P2.2 during execution of an instruction that takes 5 clock cycles to execute. P1 is received in the 2nd clock cycle of the instruction execution, P2 is received in the 3rd clock cycle, and WDT in the 4th clock cycle. Which ISR is accepted and processed first, assuming that WDTIE=1, P1.IE=0x80, P2.IE=0x04, and GIE=1? Hint: Inspect the MSP430F5529 IVT for determining priorities.

For the following questions assume that the interrupt vector table has 16 entries (0 – 15).

C. (3 points) What is the address range of the interrupt vector table (start-end)? _____

D. (3 points) What is the address of the interrupt vector with the entry number 3?

_____

E. (2 points) How does the interrupt vector table get initialized and when?

Problem #2. (25 points) Interrupts
An MSP430-based system interfaces 4 external devices (ED0, ED1, ED2, ED3), each capable of generating an interrupt request. The external devices place a request by setting the request line (a transition from a logic one to a logic zero). The request lines are connected to port 1 pins P1.3 (ED0) and P1.4 (ED1), and port 2 pins P2.3 (ED2) and P2.4 (ED3). A request line is kept active as long as the interrupt request is pending, until the request is serviced. Answer the following questions.

A (5 points) Specify the registers that need to be initialized at the beginning to configure the system for accepting the interrupts from the devices ED0-ED3. Fill in the table below. Note: to specify interrupts active on the falling edge the edge-selection bits should be set to 1.

Register	Full Name	Content after initialization [binary contnet B7… B0], b – unknown (unchanged)

B. (10 points) How many ISRs are needed to process interrupts from ED0-ED3. Outline the service routine (or routines if you need multiple ones) under the following conditions. If multiple requests occur at the same time, ED0 should have the highest priority and ED3 the lowest priority. Once in the service routine, you could service more than one pending request, but the highest priority one is serviced first.

C (10 points) Assume that processing request from each peripheral takes ~3 ms. The processor is in a low-power mode when not executing service routines. The following table describes a sequence of events in time. Fill in the table by answering what happens with the MSP430 on each relevant event if we know the following: ED2 raises an interrupt request at 13 ms, ED3 at 14 ms, and ED1 at 18 ms, and ED0 at 20 ms. The number of rows does not reflect the final solution.

Time	**Event**	**MSP430 status**
0 ms	SW initialization	Active (run initialization software)
10 ms	Go to a low-power mode	Sleep
13 ms	Request from ED2 is received, pending&accepted	P2_ISR entered (ED2 portion executed, 3 ms to go)

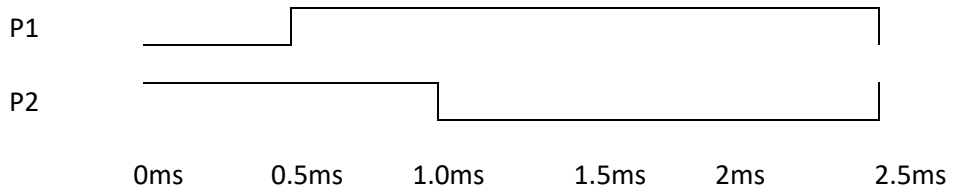
Problem 3. (25 points, TimerB, Watchdog Timer)
Consider the following code segment that utilizes the watchdog timer in the interval mode with a period set in line 4 of the code.

```
1.  #include <msp430xG46x.h>
2.  void main(void) {
3.    int p = 0;
4.    WDTCTL = WDT_ADLY_250; // check the meaning of this constant in the include file
5.    P2DIR |= BIT2;         // Set P2.2 to output direction
6.    P2OUT &= ~BIT2;        // ?
7.    for (;;) {
8.      if ((IFG1 & WDTIFG) == 1) {
9.        p++;
10.       IFG1 &= ~WDTIFG;
11.       if (p == 7) P2OUT ^= BIT2;
12.       if (p == 15) { P2OUT ^= BIT2; p=0;}
13.     }
14.    }
15. }
```

A. (5 points) What does the code segment do? What does the code in line 10 do?

B. (10 points) How would you implement the given functionality using an interrupt service routine.

C. (10 points) You would like to generate two periodic pulse-width modulated (PWM) signals P1 and P2, with frequency of 400 Hz (one period is 2.5 ms). Assume that an 2^20 Hz clock on SMCLK is used by TimerB. Can you do this using TimerB? If yes, describe a TimerB configuration (content of control and data registers) that will carry out signal generation? Note: use English and waveforms to describe your solution.



Problem 4. (20 points), Clocks Time, Timers

Consider the following code segment. Assume that processor clock in the active mode is set to 2,000,000 Hz.

```
1.  while(1) {
2.    int i;
3.    for(i = 800; i>0; i--);           // one loop iteration takes 5 clock cycles
4.    P3OUT |= BIT5;                      // Set P3.5
5.    for (i = 800; i>0; i--);           // Delay
6.    P3OUT &= ~BIT5;                     // Clear P3.5
7.  }
```

A. (5 points) What does the code segment do assuming that P3.5 is configured as a digital output. You may ignore delay needed to execute instructions in lines 1, 4 and 6.

B. (5 points) What will happen if you connect P3.5 to the buzzer?

C. (10 points) How would you implement functionality achieved by the code segment above using TimerB. Port P3.5 is multiplexed with the output signal from the capture and compare block 4 of TimerB. Give details. How would you initialize the system? What would you do in the main loop? Assume the SMCLK is used which is 2^20 =1,048,576 Hz.

Problem 5. (15 points) UART Serial Communication
Consider the following C source code (assume that P5.BIT1 is connected to a LED).

```
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCIA0RX_ISR (void)
{
  P5OUT |= BIT1;
  while(!(IFG2&UCA0TXIFG));
  UCA0TXBUF = UCA0RXBUF;
  while(!(IFG2&UCA0TXIFG));
  UCA0TXBUF = UCA0RXBUF;
  while(!(IFG2&UCA0TXIFG));
  UCA0TXBUF = UCA0RXBUF;
  P5OUT &= ~BIT1;
}
```

A. (5 points) What does this code segment do? USCIA0 is configured in the UART mode.

B. (5 points) USCI0 is configured in the UART mode to transfer 57,600 bits/second, 8-bit characters, odd parity, and two stop bits. How many processor clock cycles (MCLK = 2^20 Hz) expire during the time USCI needs to send one character over the UART?

C. (5 points) Describe how to configure USCI to achieve the desired communication speed from part B using the low frequency UART mode and SMCLK as the source clock, SMCLK=MCLK, (specify values of all important control registers and individual bit fields in these registers).