FALL 2025 EEL4742 – *Embedded Systems* - Quiz 4 Solution - Time: 30 Minutes

1. (10 points) Assume that the registers `ADC12CTL0`, `ADC12CTL1`, and `ADC12CTL2` are configured according to the specifications below. Determine the values of these registers **at conversion time in hexadecimal format**.

   [NOTE] Only consider the bits specified in the configuration below. All other bits should be assumed to be 0 (default state).

   (a) The sampling time for channel 5 is 666.66 μs, and the clock frequency is 4.8 MHz (MODOSC clock source).

   (b) The sampling time for channel 15 is 2.66 ms, and the clock frequency is 4.8 MHz (MODOSC clock source).

   (c) The pre-divider is set to 4. If an additional divider is required, use the smallest possible divider.

ADC12CTL0

15	14	13	12	11	10	9	8
ADC12SHT1x				ADC12SHT0x			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12MSC	Reserved		ADC12ON	Reserved		ADC12ENC	ADC12SC
rw-(0)	r-0	r-0	rw-(0)	r-0	r-0	rw-(0)	rw-(0)

Can be modified only when ADC12ENC = 0.

ADC12CTL1

15	14	13	12	11	10	9	8
Reserved	ADC12PDIV		ADC12SHSx			ADC12SHP	ADC12ISSH
r-0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12DIVx			ADC12SSELx		ADC12CONSEQx		ADC12BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

Can be modified only when ADC12ENC = 0.

ADC12CTL2

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
Reserved		ADC12RES		ADC12DF	Reserved		ADC12PWRMD
r0	r0	rw-(1)	rw-(0)	rw-(0)	r0	r0	rw-(0)

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved. Always reads as 0.
14-13	ADC12PDIV	RW	0h	ADC12_B predivider. This bit predivides the selected ADC12_B 00b = Predivide by 1 01b = Predivide by 4 10b = Predivide by 32 11b = Predivide by 64
8	ADC12ISSH	RW	0h	ADC12_B invert signal sample-and-hold. 0b = The sample-input signal is not inverted. 1b = The sample-input signal is inverted.

Bit	Field	Type	Reset	Description
7-5	ADC12DIVx	RW	0h	ADC12_B clock divider 000b = /1 001b = /2 010b = /3 011b = /4 100b = /5 101b = /6 110b = /7 111b = /8
4-3	ADC12SSELx	RW	0h	ADC12_B clock source select 00b = ADC12OSC (MODOSC) 01b = ACLK 10b = MCLK 11b = SMCLK

Field	Type	Reset	Description
ADC12MSC	RW	0	ADC12_B multiple sample and conversion. Valid only for sequence or repeated modes. Can be modified only when ADC12ENC = 0. 0b = The sampling timer requires a rising edge of the SHI signal to trigger each sample-and-convert. 1b = The incidence of the first rising edge of the SHI signal triggers the sampling timer, but further sample-and-conversions are performed automatically as soon as the prior conversion is completed.
Reserved	R	0	Reserved. Always reads as 0.
ADC12ON	RW	0	ADC12_B on. Can be modified only when ADC12ENC = 0. 0b = ADC12_B off 1b = ADC12_B on
Reserved	R	0	Reserved. Always reads as 0.
ADC12ENC	RW	0	ADC12_B enable conversion. 0b = ADC12_B disabled 1b = ADC12_B enabled
ADC12SC	RW	0	ADC12_B start conversion. Software-controlled sample-and-conversion start. ADC12SC and ADC12ENC may be set together with one instruction. ADC12SC is reset automatically. 0b = No sample-and-conversion-start 1b = Start sample-and-conversion

SHT1 channels 8-23

0000b = 4 ADC12CLK cycles
0001b = 8 ADC12CLK cycles
0010b = 16 ADC12CLK cycles
0011b = 32 ADC12CLK cycles
0100b = 64 ADC12CLK cycles
0101b = 96 ADC12CLK cycles
0110b = 128 ADC12CLK cycles
0111b = 192 ADC12CLK cycles
1000b = 256 ADC12CLK cycles
1001b = 384 ADC12CLK cycles
1010b = 512 ADC12CLK cycles
1011b = Reserved
1100b = Reserved
1101b = Reserved
1110b = Reserved
1111b = Reserved

SHT0 channels 0-7 & 24-31

0000b = 4 ADC12CLK cycles
0001b = 8 ADC12CLK cycles
0010b = 16 ADC12CLK cycles
0011b = 32 ADC12CLK cycles
0100b = 64 ADC12CLK cycles
0101b = 96 ADC12CLK cycles
0110b = 128 ADC12CLK cycles
0111b = 192 ADC12CLK cycles
1000b = 256 ADC12CLK cycles
1001b = 384 ADC12CLK cycles
1010b = 512 ADC12CLK cycles
1011b = Reserved
1100b = Reserved
1101b = Reserved
1110b = Reserved
1111b = Reserved

Bit	Field	Type	Reset	Description
15-6	Reserved	R	0h	Reserved. Always reads as 0.
5-4	ADC12RES	RW	2h	ADC12_B resolution. This bit defines the conversion result resolution. This bit should only be modified when ADC12ENC=0. 00b = 8 bit (10 clock cycle conversion time) 01b = 10 bit (12 clock cycle conversion time) 10b = 12 bit (14 clock cycle conversion time) 11b = Reserved
3	ADC12DF	RW	0h	ADC12_B data read-back format. Data is always stored in the binary unsigned format. 0b = Binary unsigned. 1b = Signed binary (2s complement), left aligned.
0	ADC12PWRMD	RW	0h	Enables ADC low-power mode for ADC12CLK with 1/4 the specified maximum for ADC12PWRMD = 0. This bit should only be modified when ADC12ENC = 0. 0b = Regular power mode where sample rate is not restricted 1b = Low power mode enable, ADC12CLK can not be greater than 1/4 the device-specific data sheet specified maximum for ADC12PWRMD = 0

Answer: For channel 5, with sampling time = 666.66 μs and the clock frequency = 4.8 MHz, the number of clock cycles is: $666.66 \times 10^{-6} \times 4.8 \times 10^{6} \simeq 3200$. The pre-divider is set to 4, meaning that $3200/4 = 800$ clock cycles will be needed. As the SHT maximum value is 512 (which is less than the required 800 clock cycles), an additional divider is required. The smallest divider that works here is divide-by-2. Then, $800/2 = 400$ clock cycles (can be mapped to 1010b, which corresponds to 512 ADC12CLK cycles).

For channel 25, the sampling time is 2.66 ms. Therefore, $2666.66 \times 10^{-6} \times 4.8 \times 10^{6} \simeq 12800$. Dividing by 4 gives $12800/4 = 3200$. To make it fit within the 512-cycle range, it should be divided by 7. Since the divider is shared, /7 must be used for channel 5 as well. Then, with $800/7 = 114$, SHT0 can be mapped to 0110b. For channel 25, $3200/7 = 457$, meaning SHT1 should be 1010b (512 ADC12CLK cycles).

With the other configurations, the register values at conversion time are as follows:

`ADC12CTL0` = 1010b 0110b 1001b 0011b = 16'hA693

`ADC12CTL1` = 0010b 0000b 1100b 0001b = 16'h20C1

`ADC12CTL2` = 0000b 0000b 0000b 0000b = 16'h0000

2. (5 points) **Bonus**

The MSP430 uses the ADC12_B module to read two analog channels (A4 and A10) in sequence. The goal is to compute the difference between them and send it over UART every 0.8 s. Below is the main() function. Fill in the missing lines to complete the program logic.

```
int main(void) {
    WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;              // Unlock GPIOs

    Initialize_UART();
    Initialize_ADC();

    uint16_t a4, a10;
    int16_t diff;

    while (1) {
        ________;
        while (_________);

        a4  = _________;
        a10 = _________;
        ___________________;
        ___________________;
        ___________________;
    }
}
```

The helper functions below are available:

```
void uart_write_uint16(unsigned int data);
void __delay_cycles(int ms);
```

Answer: This program continuously measures two analog voltages (A4 and A10) using the MSP430's ADC12_B module. Inside the loop, ADC12CTL0 | = ADC12SC; starts the conversion sequence, and while (ADC12CTL1 & ADC12BUSY); waits until both channels finish converting. The results are then read from ADC12MEM0 and ADC12MEM1, which hold the A4 and A10 values. The code computes their difference, sends the absolute value through UART, and waits 0.8 seconds before repeating. In short, it reads two ADC inputs, finds their voltage difference, and reports it regularly over UART.

```
...
    while (1) {
        ADC12CTL0 |= ADC12SC;                       // Start ADC conversion
        while (ADC12CTL1 & ADC12BUSY);              // Wait until conversion done

        a4  = ADC12MEM0;                            // Read channel A4
        a10 = ADC12MEM1;                            // Read channel A10

        diff = (int16_t)a10 - (int16_t)a4;          // Compute difference
        uart_write_uint16((diff >= 0) ? diff : -diff);  // Send absolute value
        __delay_cycles(800);                      // Delay
    }
```