EE 215 Homework 7

(4 problems, 80 pts total)

1. Bit banging (6 pts)
   Write one C program line to complete each task below. Explain the bit mask and show your work for bit wise operations.

   a. Clear bit 3 of the register P2OUT, but keep other bits unchanged

   b. Set bit 5 of register TA0CTL, but keep other bits unchanged

   c. Toggle bit 4 of register P1OUT, but keep other bits unchanged

Answers:

a. P2OUT &= ~BIT3;

Explanation: This line uses a bitwise AND operation with the complement of a bit mask to clear bit 3 of P2OUT, keeping other bits unchanged. The bit mask is created by left-shifting 1 by 3 positions and then complementing it.

```
            P2OUT  XXXX XXXX
            ~BIT3  1111 0111
       ---------------------
P2OUT & ~BIT3      XXXX 0XXX
```

b. TA0CTL |= BIT5;

Explanation: This line uses a bitwise OR operation with a bit mask to set bit 5 of TA0CTL, keeping other bits unchanged. The bit mask is created by left-shifting 1 by 5 positions.

```
          TA0CTL = XXXX XXXX
            BIT5 = 0010 0000
     ------------------------
TA0CTL | BIT5    = XX1X XXXX
```

c. P1OUT ^=BIT4;

Explanation: This line uses a bitwise XOR operation with a bit mask to toggle bit 4 of P1OUT, keeping other bits unchanged. The bit mask is created by left-shifting 1 by 4 positions.

```
        P1OUT  XXX1(0)  XXXX
         BIT4  000 1    0000
   --------------------------
P1OUT  ^BIT4   XXX0(1)  XXXX
```

2. Timer A (34 pts)
   Explain the following words. If it is a register, explain all bits. If it is a substitute for bits (macro), write out the binary and hex equivalent and explain pin/clock connections. [x4xx Family User's Guide, x5xx Family User's Guide]

   TA0CTL
   TAIFG
   TA0CCR0
   MC_1
   ID_2
   TASSEL_1
   TAIE
   INCLK
   ACLK
   SMCLK
   P2IN
   P2OUT
   P2DIR
   P2REN
   BIT0
   BIT3
   BIT7

Answers:

TA0CTL:
Timer A0 Control Register, controlling various aspects of Timer A0 operation.
bit 15-10 reserved
bit 9-8 TASSEL, timer A source select one of 4 clock sources
bit 7-6 input divider for the clock source
bit 5-4 mode control: stop, up, continuous, up/down
bit 3 reserved
bit 2 TACLR resets the TAxR timer value to zero
bit 1 TAIE flag to enable interrupt request in bit 0
bit 0 Timer A interrupt flag TAIFG
TAIFG:
a hex equivalent, 0b0000000000000001, 0x0001
BIT0 of TAXCTL,
Timer A interrupt flag.
Set in UP mode, the timer reaches the value in the register TA0CCR0 and
Reset to zero in TAR
TA0CCR0:
16-bit registers
Comparison mode: TAxCCRn saves data compared to the timer value in the Timer_A register: TAR.
Capture Mode: Copies the Timer_A register TAR into the TAxCCRn register
MC_1:

0b0000000000010000 , 0x0010, a hex equivalent
Mode Control 1, a macro representing "Up Mode" for Timer A.
ID_2:
0b0000000010000000, 0x0080, a hex equivalent
Input Divider 2, a macro representing a division factor of 4.
TASSEL_1:
0b0000000100000000, 0x0100, a hex equivalent
Timer A clock source select, a macro representing ACLK (ACLK = external clock).
TAIE:
0b0000000000000010, 0x0002, a hex equivalent
Timer A interrupt enable.
INCLK:
0b0000001100000000, 0x0300, a hex equivalent
Internal Clock. Timer A clock source select: 3 - INCLK
ACLK:
0b0000000100000000, 0x0100, a hex equivalent
Selects Aux Clock.
SMCLK:
0b0000001000000000, 0x0200, a hex equivalent
selects the Sub Main Clock
P2IN
Port 2 Input,
P2OUT
Port 2 Output,
P2DIR
Port 2 Direction.   enables the pullup/pulldown resistor
P2REN:
Port 2 Enable registers.
BIT0
a hex equivalent
0x01 , 0b0000 0001
BIT3
a hex equivalent
0x08 , 0b0000 1000
BIT7
a hex equivalent
0x08 , 0b0000 1000

3. Clocks (10 pts)
   Explain the difference between the three clocks of the MSP430, and list their frequency of operation.

Answers:

The master clock MCLK is used by the CPU and some peripherals. Typically 0.8 to 1.1 MHz from DCO. The 5529 has a 25 MHz master clock. (DCO = Numeric way.)Controlled Oscillator)

The subsystem master clock, SMCLK, is assigned to the peripherals. Typically 0.8 to1.1MHz from DCO.
The auxiliary clock ACLK is also assigned to the peripherals. Its source is low. Frequency crystal oscillator, typical 32.768 kHz..

4. C program, Timer A (30 pts)
   Write a C program for the MSP430. LED1 (P1.0) should initially be off.
   - If S1 (P2.6) is pushed, blink the LED at ½ second intervals, using Timer A.
   - If S1 is pushed again the cycle of 'off' and 'blinking' should repeat.

Answers:

```
include <msp430.h>

unsigned int state, debounce;

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    P1OUT &= ~BIT0; // LED1 off
    P1DIR |= BIT0;  // Set LED1 as output

    P2DIR &= ~BIT6; // Input direction for S1
    P2REN |= BIT6;  // Input resistor enable for S1
    P2OUT |= BIT6;  // Pull-up resistor for S1

    TA0CTL |= TASSEL_1 | MC_0 | TACLR | ID_0; // ACLK, count up to CCR,
divide clock by 1
    TA0CCR0 = 32768; // 1 second

    state = 0; // LED1 off is state=0, LED1 blinking is state=1
    debounce = 0;

    while(1) {
        if ((P2IN & BIT6) == 0 && debounce == 0) {
            __delay_cycles(5000); // Debounce delay
            debounce = 1;

            if (state == 0) {
                TA0CTL |= MC_1; // Turn on TimerA0
                state = 1;      // LED1 blinking
            } else {
                TA0CTL &= ~BIT4; // Turn off TimerA0
                P1OUT &= ~BIT0;  // Turn off LED1
                state = 0;       // LED1 state is not blinking
            }
        }

        if ((P2IN & BIT6) == BIT6 && debounce == 1) {
            debounce = 0;
        }

        if (TA0CTL & TAIFG) { // If flag is set from counting up
            P1OUT ^= BIT0;      // Toggle LED1
            TA0CTL &= ~BIT0;    // Reset Flag
        }
    }
} // end main
```