Hw-1 Solutions

Q1.

Part a.

```
// Set bit 5
data |= BIT5;  // BIT5 = 0010 0000

// Clear bit 5
data &= ~BIT5;  // BIT5 = 0010 0000

// Invert bit 5
data ^= BIT5;  // BIT5 = 0010 0000
```

Part b.

```
// Set bits 2 and 3
data |= (BIT2 | BIT3);  // BIT2 = 0000 0100, BIT3 = 0000 1000

// Clear bits 2 and 3
data &= ~(BIT2 | BIT3);  // BIT2 = 0000 0100, BIT3 = 0000 1000

// Invert bits 2 and 3
data ^= (BIT2 | BIT3);  // BIT2 = 0000 0100, BIT3 = 0000 1000

// Set bit 2 and clear bit 3
data = (data | BIT2) & ~BIT3;  // BIT2 = 0000 0100, BIT3 = 0000 1000
```

Part c.

```c
// Check if bit 4 is 1
if (data & BIT4) { /* bit 4 is 1 */ }  // BIT4 = 0001 0000

// Check if bit 4 is 0
if (!(data & BIT4)) { /* bit 4 is 0 */ }  // BIT4 = 0001 0000

// Check if bits 4 and 5 are 1,1
if ((data & (BIT4 | BIT5)) == (BIT4 | BIT5)) { /* bits 4 and 5 are 1,1 */ }

// Check if bit 4 is 0 and bit 5 is 1
if (!(data & BIT4) && (data & BIT5)) { /* bit 4 is 0 and bit 5 is 1 */ }

// Check if bits 4 and 5 are 0,0
if (!(data & (BIT4 | BIT5))) { /* bits 4 and 5 are 0,0 */ }
```

Q2.

Part a.

```c
// Configure the module: (Sleep mode 3)(Clock speed 4)(Capacitor value 1)(Interrupts enabled)
uint8_t CTL = SLP_3 | CLK_4 | CAP_1 | IE;
```

Part b.

- **Masks used**:
- SLP_3 = 1100 0000
- CLK_4 = 0010 0000
- CAP_1 = 0000 0010
- IE = 0000 0001
- **Final value of CTL**:
- 11100011 (in binary)

Part c.

```
// Clear the SLP field and set it to SLP_1 (0100 0000)

CTL = (CTL & ~SLP_3) | SLP_1;  // SLP_3 = 1100 0000, SLP_1 = 0100 0000
```

Part d.

```
// Check if SLP field is equal to SLP_3 (1100 0000)

if ((CTL & SLP_3) == SLP_3) {  // SLP_3 = 1100 0000
```

Part e.

```
// Check if CLK is 0, 2, 4, or 6

if ((CTL & CLK_7) == CLK_0 || (CTL & CLK_7) == CLK_2 ||
   (CTL & CLK_7) == CLK_4 || (CTL & CLK_7) == CLK_6)
{    // CLK is 0, 2, 4, or 6
}
```

Q3.

Part a.

An 18-bit address means 2^18 addresses.

Each address corresponds to 1 byte, so the total memory size is 2^18 bytes = 262144 bytes = 256KB

Part b.

To address 17,408 bytes, we need to find the smallest number of address bits that can cover it.

2^14=16,384 bytes and 2^15=32,7682 bytes. The smallest address size is 15 bits (as 14 bits would not cover 17,408 bytes).

Q4.

Part a)

Address range is 0x0500 to 0x0CFF, which means the size is:
0x0CFF−0x0500+1=0x07FF+1=2048 bytes

Part b)

Address range is 0xFFC0 to 0xFFFF, so the number of vectors is:
0xFFFF−0xFFC0+1=0x003F+1=64 vectors