#HW, cec32x_HW_22_HW2_embedded_C, 110 points total.

## HW 2. Embedded C programming

Prob 2-1. (70 points total, 5 points each) Read the code below and answer the questions for `C[0]` to ’`C[13]`:

```
0x5 = 0b0101
0x6 = 0b0110
0x7 = 0b0111
0x9 = 0b1001
0xA = 0b1010
0xB = 0b1011
```

```
int main(void) {
    uint16_t A[4] = {0x5555, 0x6666, 0x9999, 0xAAAA};
    uint16_t B[4] = {0x6666, 0x7777, 0xAAAA, 0xBBBB};
    uint16_t C[20];
    uint16_t Mask = 7;        // a mask for 3 bits
    uint16_t NUM_of_bits_to_shift = 4;
    uint16_t VALUE_to_assign = 5;
    uint16_t *pInt;

    //printf("The address of array C: %x\n", C);
    printf("Please determine the values in Hexadecimal of C:\n");
    pInt = &A[1];
    C[0] = A[0] & B[0];
    C[1] = A[1] | B[1];
    C[2] = A[2] && B[2];
    C[3] = A[3] ^ B[3];
    C[4] = 15 % 4;
    C[5] = Mask << NUM_of_bits_to_shift;
    C[6] = ~(Mask << NUM_of_bits_to_shift);
    C[7] = A[0] & ~(Mask << NUM_of_bits_to_shift);
    C[8] = (VALUE_to_assign << NUM_of_bits_to_shift);
    C[9] = A[1] | (VALUE_to_assign << NUM_of_bits_to_shift);
    C[10] = *(pInt + 2);
    C[11] = *pInt++;     pInt = &A[2] after this statement
    C[12] = (*pInt)++;
    C[13] = *pInt;
```

```
C[0]:  0b0101                  C[1]:  0b0110
     & 0b0110                       | 0b0111
-------------                  -------------
       0b0100 = 0x4                   0b0111 = 0x7
=> C[0] = 0x4444               => C[1] = 0x7777

C[2]:  0b1                C[3]:  0b1010              C[4] = 3
     & 0b1                     ^ 0b1011              C[5] = 0x0070
-------------             -------------              C[6] = 0xFF8F
       0b1                       0b0001 = 0x1
=> C[2] = 0x0001          => C[3] = 0x1111

C[7]:  0b0101             C[8] = 0x0050              C[9]:  0b0110
     & 0b1000                                             | 0b0101
-------------                                        -------------
       0b0000 = 0x0                                         0b0111 = 0x7
=> C[7] = 0x5505                                     => C[9] = 0x6676

                                                     C[10] = 0xAAAA
                                                     C[11] = 0x6666
                                                     C[12] = 0x9999
                                                     C[13] = 0x999A
```

Soln to Prob 2-2. a. 0x1F = 0b0000_0000_0001_1111. b. 0x1234 = 0b0001_0010_0011_0100

Prob 2-2 (10 points, 5 each) Convert the following hexadecimal numbers to 16-bit binary number. (Example `0xFF = 0b0000_0000_1111_1111`) a. `0x1F`, b. `0x1234`

Soln to Prob 4. 2^{12} - 1 = 4096 - 1 = 4095.

Prob 2-3 (10 points) Consider a 12-bit ADC (analog to digital converter), which converts a voltage from 0 to 5 V to a digital number between 0 and the maximum unsigned number that a 12-bit system can express. What is that maximum number?

Prob 2-4 (20 points) Determine the exact address of the IDR for GPIO port B according to the code on pages 7 and 8 of the class notes for Embedded C.

IDR's address in the data structure is 0x10, and GPIOB's address is:

- AHB2PERIPH_BASE = 0x4000_0000 + 0x0800_0000 = 0x4800_0000
- AHB2PERIPH_BASE + 0x0400 = 0x4800_0400

=> Address of (GPIOB->IDR) = 0x4800_0410.