

# EECS 373 Winter 2015 Homework #1b

---

Due January 20<sup>th</sup> at the start of class.

Name: \_\_\_\_\_ username: \_\_\_\_\_

You are to turn in this sheet as a cover page for your assignment. The rest of the assignment should be stapled to this page. *Assignments that are unstapled, lack a cover sheet, or are difficult to read will lose at least 50% of the possible points and we may not grade them at all.* This is an individual assignment; all work should be your own.

---

1. Using the ARMv7-M Architecture Reference Manual.
  - a. In straightforward English, explain what the LSR (register) instruction does. **[3 points]**
  - b. Write the hexadecimal for the machine code you would expect to get for the following instructions. **[4 points, 2 each]**
    - i. `LSLS R2, R5, #27`
    - ii. `LSLS R2, R5`
  - c. Explain what is wrong with this instruction. Write a replacement instruction that does the same thing and the hexadecimal machine code that your replacement instruction would generate. **[3 points]**
    - i. `LSLS R11, R3, #13`
  
2. For each of the following program segments, assume you start with all memory locations in question equal to zero. Indicate the values found in these memory locations when the programs end. Write all answers in hex. **[10 points, 5 for each part]**
  - a. 

```
BASE_EMC = 0xA5000000;
uint32_t *x = (uint32_t*)BASE_EMC;
*x = 0xfedcba98;
*(x-1) = 0x01234567;
```

Base Address	00	01	02	03
0xA5000004				
0xA5000000				
0xA5FFFFFFC				
0xA5FFFFFF8				

```

mov r1, #0x100
movw r2, #136
movt r2, #8
strh r2, [r1], #-3
str r2, [r1, #2]!

```

Address	Value
0FE	
0FF	
100	
101	
102	
103	
104	
105	

Hint: The ARMv7-M Architecture Reference Manual may be useful here.

3. Write a program in UAL assembly that does the same thing as the following C code. You should assume “print” is an ABI compliant function that takes a single integer argument and does something with it. Have the main return to whatever called it just as any function might. Do not use IT statements or conditional instructions (e.g. ADDNE). Annotate each line of code with what it is doing. **[10 points]**

```

int main() {
    int i, a=1;
    for(i=0; i<9; i+=2) {
        a = a + i;
        print(a);
    }
    return a;
}

```

4. Write a program in C that does the same thing as the following ARM assembly language code. Your C code must not exceed three lines. **[10 Points]**

```

movw r0, #0x1008
movt r0, #0xA500
ldr r1, [r0]
sub r1, r1, #5
str r1, [r0]

```

5. Write an ABI compliant function using the fewest instructions possible named “mean” which takes four integer arguments and returns the average. Do not worry about fractions or overflow. **[10 points]**