EECS 373 - Homework #1

Name: **unique name:**

Due 25 January via Gradescope. Please use the answer boxes provided. Submit a PDF of your completed assignment to Gradescope. Typed answers and neat handwritten answers are both acceptable.

Question 1: Short answer questions: **[10 points, 2 each]**

A) What type of memory is executable code typically stored in, when it must survive power loss?

B) What is the memory range for the peripheral devices, based on Slide 22 of Lecture 2?

C) Is the ARM ISA and hardware capable of supporting Big Endian addressing?

D) Using at most one sentence, indicate the main difference between the ARM sub and subs instructions.

E) Is an ABI part of an ISA?

Question 2

Part A:
Using the ARMv7-M Architecture Reference Manual describe in a straightforward manner what the ROR (immediate) instruction does. **[3 points]**

Part B:
Write the hexadecimal for the machine code you would expect to get for the following instructions. **[9 points, 3 each]**

1) LSL R1, R4, #4

2) LSR R1, R2, #24

3) ASR R12, R3, #1

Question 3

For each of the following program segments, assume you start with all memory locations equal to zero. Indicate the values found in *these* memory locations when the programs end. Write all answers in hex. **[16 points, 8 for each part]**

Part A)

```
BASE_EMC = 0x74000000;
uint32_t *a = (uint32_t*)BASE_EMC;
*a = 0x01234567;
*(a-1) = 0xfedcba98;
*(uint32_t*)((uint32_t)a+2)=0x01234567;
```

Address	Value
0x73FFFFFD	
0x73FFFFFE	
0x73FFFFFF	
0x74000000	
0x74000001	
0x74000002	
0x74000003	
0x74000004	

Part B)

```
mov r2, #100
movw r1, #85
movt r1, #85
strh r1, [r2, #3]
str r1, [r2], #2
strb r1,[r2, #2]!
strb r2,[r2, #-3]
```

Address	Value
100	
101	
102	
103	
104	
105	
106	

Question 4

Write an ABI compliant assembly function that checks if an unsigned integer has a square root that is an unsigned integer and returns the square root. For example if the function is given 25, it should return 5 (since 25=5^2). If no such unsigned integer exists, the function should return -1. **[20 points]**

Question 5

Given the C code below, write an equivalent program in assembly. You can assume that “print” is an ABI compliant function which takes an integer argument. Have the function return to the program that called it. **[17 points]**

```
void myFunc(void) {
      int i,a=401;
      for(i=0;i<8;i++) {
            a=a-i;
            print(a);
      }
}
```

Question 6

The following tables represent two sections of memory in a Cortex-M processor. For convenience the memory values in the address `0x08000XXX` block have been decoded into instructions. Assume that the Program Counter (PC) is `0x08000104` and the Stack Pointer is `0x20000000`. Determine the value of R0 and the values of the memory addresses in the right table (i.e. `0x20000000 to 0x1FFFFFD0`) when the PC = `0x0800010C`. Leave any unknown memory values blank. **[25 points]**

Address	Instruction	
`0x08000104`		`MOV R0, #5`
`0x08000108`		`BL` *func*
`0x0800010C`	*done*	`B` *done*
`0x08000110`	*func*	`PUSH {R4, LR}`
`0x08000114`		`MOV R4, R0`
`0x08000118`		`CMP R4, #1`
`0x0800011C`		`BNE` *else*
`0x08000120`		`MOV R0, #1`
`0x08000124`	*loop*	`POP {R4, PC}`
`0x08000128`	*else*	`SUB R0, R4, #1`
`0x0800012C`		`BL` *func*
`0x08000130`		`MUL R0, R4, R0`
`0x08000134`		`B` *loop*

Address	Value
`0x20000000`	
`0x1FFFFFFC`	
`0x1FFFFFF8`	
`0x1FFFFFF4`	
`0x1FFFFFF0`	
`0x1FFFFFEC`	
`0x1FFFFFE8`	
`0x1FFFFFE4`	
`0x1FFFFFE0`	
`0x1FFFFFDC`	
`0x1FFFFFD8`	
`0x1FFFFFD4`	
`0x1FFFFFD0`	
R0	