National University of Singapore
Dept of Electrical & Computer Engineering (ECE)

CG2028 Computer Organization

Tutorial 2: ARMv7E-M Assembly Language

1. Which of the following ARM instructions will cause the assembler to issue a syntax error message? Explain why?

(a)	ADD	R2, R2, R2
(b)	SUB	R0, R1, [R2, #4]
(c)	MOV	R0, R1, R2
(d)	LDR	R0, R1
(e)	ADDS	R0, R1, R2, LSL #3
(f)	BLXAL	LOL
(g)	MVNS	R1, R1
(h)	EOR	R1, R1, R1
(i)	RRX	R1, R1
(j)	TEQS	R1, R2
(k)	PUSH	{R0-R16}
(l)	POPS	{R0, R1, R2}

2. Assume the following register and memory contents in an ARM embedded system. What is the effect of executing the following instructions with the given initial values?

Main Memory	
0x00000100	**LDR R8, [R0]**
0x00000104	**LDR R9, [R0, #4]**
0x00000108	**ADD R10, R8, R9**
:	:
0x00001000	1
0x00001004	2
0x00001008	3
0x0000100C	4
0x00001010	5
0x00001014	6
:	:

Registers	
PC	
:	
R0	0x1000
R1	0x2000
R2	0x1010
R3	0x300
:	:
R8	0x400
R9	0x500
R10	0x600
:	:

3. Assume the following register and memory contents in an ARM embedded system. What is the effect of executing the following instructions with the given initial values?

Main Memory	
0x00000100	**STR R6, [R1, #-4]!**
0x00000104	**STR R7, [R1, #-4]!**
0x00000108	**LDR R8, [R1], #4**
0x0000010C	**LDR R9, [R1], #4**
0x00000110	**SUB R10, R8, R9**
:	:
0x00001000	1
0x00001004	2
0x00001008	3
0x0000100C	4
0x00001010	5
0x00001014	6

Registers	
R0	0x1000
R1	0x2000
R2	0x1010
R3	0x300
R4	0x200
R5	10
R6	20
R7	30
R8	0x400
R9	0x500
R10	0x600
:	:

4. Write an ARM assembly program that finds the number of negative integers in a list of n 32-bit integers and stores the count in location NEGNUM. The value n is stored in memory location N, and the first integer in the list is stored in location NUMBERS.
   Hint: use the IF-THEN (IT) block.

5. Write an ARM assembly program to reverse the order of the bits in register R2. For example, if the starting pattern in R2 is 1110…0100, the final result in R2 should be 0010…0111.
   Hint: use shift and rotate operations.

6. Write an ARM assembly program that generates the first n numbers of the Fibonacci series. In this series, the first two numbers are 0 and 1, and each subsequent number is generated by adding the preceding two numbers. For example, for n=8, the series is 0, 1, 1, 2, 3, 5, 8, 13. Your program should store the numbers in successive memory word locations starting at MEMLOC. Assume that the value n is stored in location N.
   Hint: assumes that the last number in the series of n numbers can be represented in a 32-bit word, and that n >2.

END