CSC111 Assembly Language Programming

Fall 2025 Midterm Exam

Student Name: _____ID：_____s

Total Points	

Note: This exam paper should be kept confidential and any dissemination violates copyright.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
/10	/10	/20	/10	/10	/10	/20	/10

(In all the problems, the system is assumed to be 32-bit unless specified otherwise.)

Q1 Multiple-choice questions: enter your answer keys here:

1	2	3	4	5	6	7	8	9	10

For the following multiple-choice questions, each question has exactly one correct answer key. If multiple choices are correct, choose the option "All of the above". **Fill in the answer keys in the table above. (Answer keys written in the question area will not be counted.)**

1. Which registers are general-purpose on ARM Cortex-M?
   A. R0–R12
   B. R13–R15
   C. CONTROL, BASEPRI, PRIMASK
   D. xPSR only

2. What are the roles of R13, R14, and R15 on ARM Cortex-M?
   A. General purpose
   B. Program counter, stack pointer, and link register in that order
   C. Stack pointer (R13), link register (R14), program counter (R15)
   D. Interrupt mask registerse

3. On ARM Cortex-M3, the borrow and carry flags relation is:
   A. Carry = Borrow
   B. Carry = NOT Borrow
   C. Borrow always 0
   D. Carry always 0

4. In two's complement, TC(x) can be obtained by:
   A. Invert bits
   B. Invert bits and subtract one
   C. Invert bits and add one
   D. Add one then invert bits

5. In a 5-bit system, which statement is true about −16 (10000_2)?
   A. Its two's complement is 00000_2
   B. Its two's complement is itself
   C. It cannot be represented
   D. It equals +16

6. What is the bit width of each register in ARM Cortex-M processors?
   A) 16 bits
   B) 24 bits
   C) 32 bits
   D) 64 bits

7. In ARM assembly instruction format, what is typically the first operand (operand1)?
   A) Source register
   B) Immediate value
   C) Destination register
   D) Memory address

8. When adding two 64-bit integers split across two registers each, which instruction pair correctly handles the low and high halves?
   A. ADC for high halves, then ADDS for low halves to set carry
   B. ADC for low halves, then ADDS for high halves
   C. ADDS for low halves to set carry, then ADC for high halves
   D. ADD for low halves, then ADD for high halves

9. In ARM subtraction, what does the carry flag C indicate when a borrow occurs in SUBS?
   A. C = 1 when there is a borrow
   B. C = 0 when there is a borrow
   C. C toggles regardless of borrow
   D. C is always preserved from the previous instruction

10. Which single instruction multiplies a register by 17 using the barrel shifter on the second operand?
    A. ADD r4, r4, r4, LSL #4
    B. RSB r5, r5, r5, LSL #5
    C. ADD r1, r0, r0, ASR #3
    D. MUL r1, r0, #17

Q2. (10 points) Assuming a 4-bit system. Show the equivalent decimal values when the data is interpreted as unsigned binary or signed binary.

Binary Value	Signed Decimal Value	Unsigned Decimal Value
1111		
1011		
0110		
1001		
0011		

Q3. (20 points)

(a) (5 points) Assume an 8-bit system. Copy bits 5..3 in R4 to be the rightmost bits of R5, and set the other R5 bits to 0.

(b) (5 points) Assume an 8-bit system. Copy the bottom four bits 3..0 in R0 to bits 6..3 in R1, and keep the other bits in R1 unchanged.

(c) (10 points) Assume R0 and R1 are initialized with the values below. Write a sequence of assembly instructions that would produce the values in registers R2–R5, from R0 and R1.

Given:

R0	0xCAFEFADE
R1	0x00FF00FF

Produce:

R2	0xCAFFFAFF
R3	0xCAFFFA8F
R4	0xFFFFFFCA
R5	0xDEFF00FF

Q4. (10 points) Assume a 4-bit system. For each of the following operations, compute the result and NZCV flags, based on the first row that has been given to you. s

Operation	Result in binary	Equivalent unsigned arithmetic in decimal	Equivalent signed arithmetic in decimal	NZCV
1001 + 0010	1011	9 + 2 = 11	-7 + 2 = -5	1000
1101 + 1100				
1101 - 1100				
1100 + 1010				
0100 - 0110				
0100 + 0010				

Q5. (10 points) For each of the following instructions (executed individually, not sequentially), compute the result and NZCV flags. Assume initially R0=0xFFFFFFFF, R1=0x00000001.

Instruction	Result in dest. register R0	NZCV
MOVS R0, #0		
ANDS R0, #0		
ORRS R0, R0, R1		
ANDS R0, R0, R1		
ADDS R0, R0, R1		
SUBS R0, R0, R1		
ADDS R0, R0, R1, LSL #31		
BICS R0, R0, R1, LSL #31		
EORS R0, R0, R1, LSL #31		
LSLS R0, R0, #31		

Q6. (10 points)

(a) Suppose r0 = 0x00008000, and the following memory layout:

Address	Data
0x00008007	0x79
0x00008006	0xCD
0x00008005	0xA3
0x00008004	0xFD
0x00008003	0x0D
0x00008002	0xEB
0x00008001	0x2C
0x00008000	0x1A

(a1) ARM processors can be configured as big-endian or little-endian. What is the value of r1 after running LDR r1, [r0]?

a. If little-endian, r1 = _____
b. If big-endian, r1= _____

(a2) Suppose the system is little-endian. What are the values of r1 and r0? Each instruction is executed individually, not sequentially.

a. LDR r1,[r0,#4]
b. LDR r1,[r0],#4
c. LDR r1,[r0,#4]!

After LDR r1,[r0,#4]

r0 = _____

r1 = _____

After LDR r1,[r0],#4

r0 = _____

r1 = _____

After LDR r1,[r0,#4]!

r0 = _____

r1 = _____

(b) Assume little-endian memory ordering. Suppose r0 = 0x2000,0000 and r1 = 0x12345678. All bytes in memory are initialized to 0x00. Fill the following table after the assembly program has finished execution.

```
STR r1, [r0], #4
STR r1, [r0, #4]!
STR r1, [r0]
```

Memory Address	Initial Memory Content	Final Memory Content
0x2000,0013	00	
0x2000,0012	00	
0x2000,0011	00	
0x2000,0010	00	
0x2000,000F	00	
0x2000,000E	00	
0x2000,000D	00	
0x2000,000C	00	
0x2000,000B	00	
0x2000,000A	00	
0x2000,0009	00	
0x2000,0008	00	
0x2000,0007	00	
0x2000,0006	00	
0x2000,0005	00	
0x2000,0004	00	
0x2000,0003	00	
0x2000,0002	00	
0x2000,0001	00	
0x2000,0000	00	

Q7. (20 points) Show all updates to registers as the assembly code shown below runs, assuming **big-endian** ordering. (Memory addresses increase from left to right in the table.) Fill in the tables of final register values and memory contents. Show the NZCV flags after execution, assuming NZCV=0000 initially.

Assembly program

LDR R1, =0x10000010
LDR R2, [R1]
LDR R3, [R1, #4]!
ASR R4, R2, #12
MOVW R5, #14
LSL R6, R5, #9
BIC R7, R2, R6
AND R8, R7, R6
STR R7, [R1, #8]

Initial memory contents:

0x10000010	FF	EF	CD	AB	00	00	CD	AB	AA	BB	CC	DD	EE	FF	11	22

Final register values:

R1		R2		R3		R4	
R5		R6		R7		R8	

Final memory contents:

0x10000010																

Final NZCV = _____

Q8. (10 points) Consider an array[] of 25 integers (each integer is 4 bytes). Assume register R2 holds the base memory address of array[]. A compiler associates variables x and y with registers R0 and R1, respectively. Translate this C program into ARM assembly language based on the provided comments..

C program	Assembly program

uint32_t x = array[5] + y;
array[6] = x * 4;
array[7] = x / 4;
array[8] = x - 10;
array[9] = x * (x - 1);
	_____; load array[5] (with 5*4 = 20 bytes offset) into
	; temporary register R3
	
	_____; x = array[5] + y
	
	_____; R3 = x * 4
	
	_____; store into array[6]
	
	_____; R3 = x / 4
	
	_____; store into array[7]
	
	_____; R3 = x – 10
	
	_____; store into array[8]
	
	_____; R4 = x – 1
	
	_____; R3 = x * (x - 1)
	
	_____; store into array[9]

Appendix. Conversion table. (You may tear off and discard this page.)

Decimal	Binary	Hex
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF