CSC111 Assembly Language Programming

Spring 2026 Midterm Exam

Student Name: ID:

Total Points	

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

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

Q1 (10 points) 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. 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

2. Signed overflow can occur when:
   A. Adding operands with different signs
   B. Subtracting operands with the same sign
   C. Adding two negatives
   D. Subtracting a negative from a negative

3. For char str = "ARM Assembly", what must the final byte be and what is the string's size in Bytes?
   A. 0x00; 13
   B. 0x20; 12
   C. 0x41; 13
   D. 0x79; 12 (ASCII hex code for lowercase 'y' is 0x79)

4. Which of the following is not a barrel shifter operation in ARM?
   A. LSL
   B. ROR
   C. RRX
   D. ROL

5. How can a rotate-left by n bits be implemented on a 32-bit value using available rotate instructions?
   A. As RRX by n
   B. As ROR by 32 − n
   C. As LSL by 32 − n
   D. As ASR by n

6. Which shift corresponds to signed division by a power of two with sign extension?
   A. LSR
   B. ASR
   C. LSL
   D. ROR

7. Assume Little Endian and r0 = 0x20008000 with memory contents: [0x20008000]=0xEF, [0x20008001]=0xCD, [0x20008002]=0xAB, [0x20008003]=0x89; what does LDRH r1, [r0] load into r1?
   A) 0x0000EFCD
   B) 0x0000CDEF
   C) 0x89ABCDEF
   D) 0x000089AB

8. What happens when you execute LDRSB r1, [r0] and the byte at the memory location has value 0xEF?
   A) r1 = 0x000000EF
   B) r1 = 0xFFFFFFEF
   C) r1 = 0xEF000000
   D) r1 = 0x0000FFEF

9. Which pattern correctly toggles bit 5 of r0 without affecting other bits?
   A. ORRS r0, r0, r4, LSL #5 with r4 = 1
   B. ANDS r0, r0, r4, LSL #5 with r4 = 1
   C. EORS r0, r0, r4, LSL #5 with r4 = 1
   D. BICS r0, r0, r4, LSL #5 with r4 = 1

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) Binary numbers

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
1000		
0111		
1101		
0000		
1110		

Q3. (20 points) Unsigned and signed arithmetic

Assume a 4-bit system. For each of the following operations, give the equivalent arithmetic calculation in decimal, and the NZCV flags, based on the first row that has been given to you. GT denotes Ground Truth value when there is a carry or overflow.

Operation	Result in binary	Equivalent unsigned arithmetic in decimal	Equivalent signed arithmetic in decimal	NZCV
0100 - 1110	0110	**4 - 14 = 6 (GT=-10)**	**4 - -2 = 6**	**0000**
0100 + 0110	1010			
1100 + 1110	1010			
1100 + 1010	0110			
0100 - 0110	1110			
1100 - 0110	0110			

Q4. (20 points) Bit manipulations

(a) (5 points) Assume an 8-bit system. Copy the top three bits 7..5 in R0 to bits 4..2 in R1, and keep all other bits in R1 unchanged.

ANS:

(b) (5 points) Assume an 8-bit system. Copy bits 5..2 in R0 to bits 3..0 in R1, and keep the upper four bits in R1 unchanged.

ANS:

(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. Do not use MOVW + MOVT). Do not use pseudo-instructions LDR to load large constants into a register.

Given:

R0	0xDEADBEEF

R1	0x0000FFFF

Produce:

R2	0xDEAD4110
R3	0x0000DEAD
R4	0xBEEFFFFF

ANS:

Q5. (20 points) Program execution tracing

Assume a 32-bit system with **big-endian** memory addressing. The following assembly program operates on memory and registers. Fill in the values of registers R1 through R7 and the final memory contents after the entire program has finished executing. Write your answers in hexadecimal format. Assume initially NZCV = 0000.

Assembly program:

```
LDR   R1, =0x20000000
LDR   R2, [R1]
LDR   R3, [R1, #4]!
LSR   R4, R2, #8
MOVW  R5, #15
LSL   R6, R5, #4
BIC   R7, R2, R6
STR   R7, [R1, #8]
```

Initial memory contents:

Addr	Content

0x20000000	12	34	56	78	9A	BC	DE	F0	11	22	33	44	55	66	77	88

ANS:

Final register values:

R1	R2	R3	R4
R5	R6	R7	

Final memory contents:

Addr	Content															
0x20000000																

Q6. (20 points) Assume a 32-bit system. Consider an array[] of 20 integers (each integer is 4 bytes). Assume register R0 holds the base memory address of array, i.e., array[0]. A compiler associates variables z and y with registers R1 and R2, respectively. Translate this C program into ARM assembly language based on the provided comments. Use R3 as a temporary register. (Note: Assume z is an unsigned integer. You may choose to use either pre-index, or pre-index with update, or post-index addressing.)

C program:

```
uint32_t z = array[2] - y;
array[^3] = z * 8;
array[^4] = z / 2;
array[^5] = z + 12;
array[^6] = z | 0x0F;
```

ANS: Assembly program:

_____; load array[2] into temporary register R3

_____; z = array[2] - y

_____; R3 = z * 8

_____; store into array[3]

_____; R3 = z / 2

_____; store into array[4]

_____; R3 = z + 12

_____; store into array[5]

_____; R3 = z | 0x0F

_____; store into array[6]