

## CSE2312 Computer Organization and Assembly Language Programming

### Test 1 Example Study Topics

1. Convert between binary, decimal, and hexadecimal unsigned and signed numbers.
2. Calculate the maximum and minimum value of an unsigned or signed integer (for any n-bit integer).
3. Know how to pass up to 4 right hand parameters to a function and how a left hand argument is returned from a function. As a simple example, where would a function, `uint32_t sum4(uint16_t a, uint16_t b, uint16_t c, uint16_t d)` that returns `a+b+c+d`, look for the values a-d and how would it return the result? How would the parameters be passed and the result be returned for the function `uint64_t add64(uint64_t x, uint64_t y)`? This is covered in the document on the class web page, but the rules you need to know for the exam are limited to the cases discussed in class.
4. Understand what the {S} suffix for an ARM v7 instruction does and why it is important. For example, understand why ADDS and ADC are used in the add32 example from class instead of ADD or ADCS.
5. For ARM instructions ADDS, SUBS, RSBS, ANDS, ORRS, EORS, MOVS, and MOVNS, understand how the zero, carry, and sign flags are affected.
6. Encode different constant values in the operand2 field of a MOV or MVN instruction. Understand what values can be encoded in the 12-bit field.
7. Know how shift operations work in ARM v7 instructions (e.g., `MOV R0, R1 ASL #16`). Specifically, make sure you are aware of how the Rm LSL/LSR/ASR #5bit\_imm and Rm LSL/LSR/ASR Rn operand2 encodings work with the MOV instruction.
8. Fully understand how BL and BX work. Specifically, make sure you can determine the value loaded into the LR and PC registers as a result of a BL or BX LR instruction if the address of the instructions are given..
9. Understand how Bcond can be used with {S} suffixed instructions to perform basic branch decisions. Know what the conditions are as given in Table 1 -6. A examples of this are the `max*()` and `isEqual()` coding questions.
10. Read and understand pages 1 -11 through 1 -20 of ARM document DDI0210B on the class webpage.
11. Be able to write assembly code similar to any of the HW#2 problem 5 questions on the exam. You will not need to write the C code on the exam to test the function.
12. Know how endianness affects how numbers are written and read from memory. Understand how the LDR, LDRH, LDRSH, LDRB, and LDRSB instructions work fully.

*Please make sure you have all relevant pages of the ARM technical reference, C calling and register convention document, class notes, and class code printed out before the exam (the sections should be obvious from the class examples and these example test questions/topics). No computers/tablets/phones are allowed during the exams as stated in the syllabus. Be sure to bring a calculator.*