ASM Lab 4

Lab Section Meeting Day	*Lab Start Date*	*Lab Due Date*
Tuesday	9/19	9/26
Friday	9/22	9/29

Labs can now be submitted through Brightspace!

1) Read pages 243-254. There is nothing to hand in for this portion.
2) Read pages 258-261. Take notes on what you need to know/do for the project, and hand them in as part of the lab.
3) Do some research on the `printf` C function and answer the following questions:
    a. What is/are the function's input(s)?
    b. What is the function's output, if any?
    c. Briefly describe what the function does.
4) Refer to the LCC Instruction Set Summary and chapter 3. Do the following tasks for each of the following instructions:
    i. Provide the instruction format.
    ii. Provide the symbolic description.
    iii. Describe, in words, what the instruction does.
    a. Both types of the `add` instruction
    b. Both type on the `and` instruction
    c. `br`
    d. `brn`
    e. `brp`
    f. `brz`
    g. `ld`
    h. `st`
    i. `bl`
    j. `blr`
    k. `ldr`
    l. `str`
    m. `not`
    n. `jmp` and `ret`
    o. `lea`
    p. `trap (halt, nl, dout)`
5) Explain what each of the following C bitwise operators do:
    a. `>>`
    b. `<<`
    c. `&`
6) Look at the i1shell.c code and explain what is happening in the following lines.
    a. `opcode = ir >> 12;`

   b. `pcoffset9 = ir <<7;`
      `pcoffset9 = imm9 = pcofffset9 >> 7;`
   c. `trapvec = ir&0x15;`
   d. `sr1 = baser = (ir & 0x01c0) >> 6;`

7) If the `ir = 1010 0000 1111 0101` (this is an arbitrary 16-bit word), what would the following fields be equal to:
   a. `opcode`
   b. `pcoffset9`
   c. `pcoffset11`
   d. `imm5`
   e. `offset6`
   f. `trapvec`
   g. `dr`
   h. `sr1`
   i. `sr2`
   j. `bit5`
   k. `bit11`

8) Complete the following segment of the i1 project. (You fill in wherever there is "…"):

```
// isolate the fields of the instruction in the ir
      opcode = ir >> 12;                  // get opcode
      pcoffset9 = ir << 7;                // left justify pcoffset9 field
      pcoffset9 = imm9 = pcoffset9 >> 7;  // sign extend and rt justify
      pcoffset11 = ...                    // left justify pcoffset11 field
      pcoffset11 = ...                    // sign extend and rt justify
      imm5 = ...                          // left justify imm5 field
      imm5 = ...                          // sign extend andd rt justify
      offset6 = ...                       // left justify offset6 field
      offset6 = ...                       // sign extend and rt justify
      trapvec = eopcode = ir & 0x1f;      // get trapvec and eopcode fields
      code = dr = sr = ...                // get code/dr/sr and rt justify
      sr1 = baser = (ir & 0x01c0) >> 6;   // get sr1/baser and rt justify
      sr2 = ...                           // get third reg field
      bit5 = ...                          // get bit 5
      bit11 = ir & 0x0800;                // get bit 11
```