Homework 3

Out: 10.6.21
Due: 10.18.21

1. [Assembly bits]
   For each of the following tasks write a LEGv8 assembly instruction(s) to perform the task.
   a) Add x1 and x7 and store the result in x5.
   b) Load an unsigned byte from the address in register x6 + 16 into register x0
   c) Branch to label EndIf if x5 is equal to x11 (assume unsigned content).
   d) Shift the content of register x9 right by 15 positions.

2. [Instruction Encoding]
   Provide the hexadecimal machine language instruction for the following LEGv8 assembly instructions.
   a) stur x9, [x10, 32]
   b) addi x9, x6, 8

3. [Instruction Decoding]
   Provide the instruction type and equivalent assembly language instruction for the following machine language instructions.
   a) 0x8B000000
   b) 0xB4016B54

4. [Bit manipulation]
   Assume the following register values: x10 = 0x00000000AAAAAAAA, x11 = 0x1234567812345678. Provide the value of x12 at the end of each of the following sequences of instructions.
   a) lsl x12, x10, 4
      orr x12, x12, x11
   b) lsr x12, x10, 3
      andi x12, x12, 0xFEF

5. [Assembly snippets]
   Write the corresponding LEGv8 code for each of the following C statements. Use a minimal number of LEGv8 assembly instructions. Assume that the variables f, g, h, i, and j are in registers x0, x1, x2, x3, and x4, respectively, and that the base address of the arrays A and B are in registers x6 and x7, respectively.
   a) f = g + (h – 5);
   b) B[8] = A[i-j];
   c) if (f == g) i=i+1 else i=i+2