6.004 Spring 2019 Tutorial Problems
L01 – Model of computing

Binary representation:

1. What is the 5-bit binary representation of the decimal number 21?

2. What is the hexadecimal representation for decimal 219 encoded as an 8-bit binary number?

3. What is the hexadecimal representation for decimal 51 encoded as a 6-bit binary number?

4. The hexadecimal representation for an 8-bit binary number is 0x9E. What is its decimal representation?

5. What is the range of integers that can be represented with a single 8-bit quantity?

6. Since the start of official pitching statistics in 1988, the highest number of pitches in a single game has been 172. Assuming that remains the upper bound on pitch count, how many bits would we need to record the pitch count for each game as a binary number?

7. Compute the sum of these two 4-bit binary numbers. Express the result in hexadecimal.

$$\begin{array}{r} 1101 \\ +\underline{0110} \end{array}$$

Assembly Language:

LW	lw *rd, offset(rs1)*	Load Word	`reg[rd] <= mem[reg[rs1] + offset]`
SW	sw *rs2, offset(rs1)*	Store Word	`mem[reg[rs1] + offset] <= reg[rs2]`
ADDI	addi *rd, rs1, constant*	Add Immediate	`reg[rd] <= reg[rs1] + constant`
BEQ	beq *rs1, rs2, label*	Branch if =	`pc <= (reg[rs1] == reg[rs2]) ? label : pc + 4`
BNE	bne *rs1, rs2, label*	Branch if ≠	`pc <= (reg[rs1] != reg[rs2]) ? label : pc + 4`
BLT	blt *rs1, rs2, label*	Branch if <	`pc <= (reg[rs1] < reg[rs2]) ? label : pc + 4`
BGE	bge *rs1, rs2, label*	Branch if ≥	`pc <= (reg[rs1] >= reg[rs2]) ? label : pc + 4`

li *rd, constant*	Load Immediate	`reg[rd] <= constant`

Compile the following expressions to RISCV assembly using the instructions above. Assume a is stored at address 0x1000, b is stored at 0x1004, and c is stored at 0x1008.

1. a = b + c;

2. if (a > b) c = 17;

3. sum = 0;
   for (i = 0; i < 10; i = i+1)
      sum += i;