1. Convert $4096_{10}$, $-2,047_{10}$, and $-2,000,000_{10}$ into 32-bit two’s complement binary numbers, respectively, and convert the following two’s complement binary numbers to be decimal numbers:

   a. 1111 1111 1111 1111 1111 1111 0000 0110$_{two}$
   b. 1111 1111 1111 1111 1111 1111 1110 1111$_{two}$
   c. 0111 1111 1111 1111 1111 1111 1110 1111$_{two}$.

2. The following MIPS instruction sequence could be used to implement a new instruction that has two register operands. Give the instruction a name and describe what it does. Note that register $t0$ is being used as a temporary.

   srl $s1$, $s1$, 1 #
   sll $t0$, $s0$, 31 # These 4 instructions accomplish “new $s0$ $s1$”
   srl $s0$, $s0$, 1 #
   or $s1$, $s1$, $t0$ #

3. The ALU supported set on less than ($slt$) using just the sign bit of the adder. Let’s try a set on less than operation using the values $-7_{10}$ and $6_{10}$. To make it simpler to follow the example, let’s limit the binary representations to 4 bits: $1001_{two}$ and $0110_{two}$.

   $1001_{two} - 0110_{two} = 1001_{two} + 1010_{two} = 0011_{two}$

   This result would suggest that $-7_{10} > 6_{10}$, which is clearly wrong. Hence we must factor in overflow in the decision. Modify the 1-bit ALU in the following figures to handle $slt$ correctly.

---

Figure 1: A 1-bit ALU that performs AND, OR, and addition on a and b or b’.
4. Add \(2.85_{10} \times 10^3\) to \(9.84_{10} \times 10^4\) and add \(3.63_{10} \times 10^4\) to \(6.87_{10} \times 10^3\), respectively, assuming that you have only three significant digits, first with guard and round digits and then without them.

5. Given the bit pattern:

\[
1010\ 1101\ 0001\ 0000\ 0000\ 0000\ 0000\ 0010
\]

what does it represent, assuming that it is

a. a two’s complement integer?
b. an unsigned integer?
c. a single precision floating-point number?
d. a MIPS instruction?