Problem 1 (20 points)

(a) 3219300464
(b) -1075666832
(c) 0011 0010 0001 1001 0000 0000 0100 0110 0100
    3 2 1 9 3 0 0 4 6 4

Problem 2 (20 points)

(a) 613566756
(b) 613566756
(c) 6.34413 × 10⁻¹⁷
(d) addiu $a0, $s2, 18724

Problem 3 (20 points)

(a) Invert every bit in the binary format of X, then add 1 to the result.
(b) The binary format of a positive integer, X is represented as:

\[ X = b_{n-1}b_{n-2}...b_1b_0 \]

in which, \( b_i \) is 0 or 1, where \( b_0 \) is the least significant bit and \( b_{n-1} \) is the most significant bit. Therefore:

\[ X = b_{n-1} \times 2^{n-1} + ... + b_i \times 2^i + ... + b_1 \times 2^1 + b_0 \times 2^0 \]
According to the definition of two’s complement, the negation of \( X \) is represented as:

\[
\bar{b}_{n-1}\bar{b}_{n-2}\ldots \bar{b}_1\bar{b}_0 + 1
\]

in which \( \bar{b}_i = 1 - b_i \). So, the value of negative \( X \) in two’s complement is:

\[
-2^{n-1} \times (1 - b_{n-1}) + 2^{n-2} \times (1 - b_{n-2}) + \ldots + 2^{1} \times (1 - b_{1}) + 2^{0} \times (1 - b_{0}) + 1
\]

Add these two numbers:

\[
X + (-X) = -2^{n-1} \times (1 - b_{n-1}) + 2^{n-1} \times b_{n-1} + 2^{n-2} + \ldots + 2^{1} + 2^{0} + 1
\]

Notice that:

\[
-2^{n-1} \times (1 - b_{n-1}) + 2^{n-1} \times b_{n-1} = -2^{n-1} + 2^{n-1} \times 2 \times b_{n-1}
\]

So for the case \( b_{n-1} = 0 \):

\[
-2^{n-1} + (2^{n-2} + 2^{n-3} + \ldots + 2^{1} + 2^{0} + 1) = -2^{n-1} + 2^{n-1} = 0
\]

And for the case \( b_{n-1} = 1 \):

\[
-2^{n-1} + 2^{n-1} \times 2 \times b_{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2^{1} + 2^{0} + 1 =
\]

\[
2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2^{1} + 2^{0} + 1
\]

which equals to:

\[
2^n
\]

Because the carry out of the most significant bit will be lost, this value is equivalent to 0. So, the two’s complement representation of negative \( X \) follows the definition of the negative number. Therefore, it is correct.

**Problem 4 (20 points)**

(a) 0000 0111 and 1111 1011

(b) If the most significant bit of the number is 0, fill the remaining higher bits with zero; if the most significant bit of the number is 1, fill the remaining higher bits with one.

(c) If the number is positive, the principle is self-evident. Let’s consider the case where the number is negative, i.e. the most significant bit is 1.

First we represent an i-bit field as:
\begin{align*}
    b_{i-1}b_{i-2}...b_1b_0 &= b_{i-1} \times -2^{i-1} + b_{i-2} \times 2^{i-2} ... b_1 \times 2^1 + b_0 \times 2^0
\end{align*}

Remember \( b_{i-1} = 1 \)

Now we represent the i-bit field in a j-bit field using the principle of sign extension described in the previous section.

\[-2^{j-1} + 2^{j-2} + 2^{j-3} ... 2^{i-1} + b_{i-2} \times 2^{i-2} ... b_1 \times 2^1 + b_0 \times 2^0\]

Now if \( X_i - X_j = 0 \) then \( X_i = X_j \):

\begin{align*}
    b_{i-1} \times -2^{i-1} + b_{i-2} \times 2^{i-2} ... b_1 \times 2^1 + b_0 \times 2^0 - (-2^{j-1} + 2^{j-2} + 2^{j-3} ... 2^{i-1})
\end{align*}

Notice all terms below i-1 cancel out, giving us:

\[-2^{i-1} - (-2^{j-1} + 2^{j-2} + 2^{j-3} ... 2^{i-1}) = -2^{i-1} + 2^{j-1} - 2^{j-2} - 2^{j-3} ... -2^{i-1}\]

We use the principle that \(-2^k + -2^k = -2^{k+1}\) for any \( k \) recursively until we arrive at:

\[2^{j-1} - 2^{j-1} = 0\]

**Problem 5 (20 points)**

(a) When transforming a positive number to the two’s complement negative number, you will first invert every bit in the number and add 1 to the result. The carry-in of the least significant bit is used to perform this addition.

(b) For the worst case scenario, there are no generate signals, in which case, the 4-bit adders still must ripple. Thus the gate delay is \( 2^*N+1 \) for the ripple adder plus the CLA unit requires \( 2^*(N/4) \) delays. But in the best case, there is a generate signal for every unit, in which case, all units can operate in parallel. Thus the delay is \( 2^*(N/4)+1+2 \) for the CLA.

(c) Simply add 4 to the subscripts for each iteration.