CSE230: Computer Organization and Assembly Language

Trey Manuszak
tmanusza@asu.edu
Arizona State University

September 28, 2020

Problem 1. Perform a multiplication of two binary numbers (multiplicand 0110 and multiplier 0110) by creating a table to show steps taken, multiplicand register value, multiplier register value and product register value for each iteration by following the steps described in the following document.

Solution.

Iteration	Step	Multiplicand Value	Multiplier Value	Product Value
0	Initial Values	0110	0110	0
1	2. sll Multiplicand by 1 3. srl Multiplier by 1	01100	011	0
2	1a. Prod = Prod + Multiplicand 2. sll Multiplicand by 1 3. srl Multiplier by 1	011000	01	0 + 01100 = 01100
3	1a. Prod = Prod + Multiplicand 2. sll Multiplicand by 1 3. srl Multiplier by 1	0110000	0	01100 + 011000 = 100100
4	2. sll Multiplicand by 1 3. srl Multiplier by 1	0110000	-	100100

Problem 2. Perform a division of two binary numbers (divide 0011 0110 by 0110) by creating a table to show steps taken, quotient register value, divisor register value and remainder register value for each iteration by following the steps described in the following document.

Solution.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial Values	0000	01100000	00110110
1	1. Rem = Rem - Div	0000	01100000	11010110
	2b. Rem <0, Rem += Div, sll Q, Q0 = 0	0000	01100000	00110110
	3. srl Div	0000	00110000	00110110
2	1. Rem = Rem - Div	0000	00110000	00000110
	2a. Rem >= 0, sll Q, Q0 = 1	0001	00110000	00000110
	3. srl Div	0001	00011000	00000110
3	1. Rem = Rem - Div	0001	00011000	11101110
	2b. Rem <0, Rem += Div, sll Q, Q0 = 0	0010	00011000	00000110
	3. srl Div	0010	00001100	00000110
4	1. Rem = Rem - Div	0010	00001100	11111010
	2b. Rem <0, Rem += Div, sll Q, Q0 = 0	0100	00001100	00000110
	3. srl Div	0100	00000110	00000110
5	1. Rem = Rem - Div	0100	00000110	00000000
	2a. Rem >= 0, sll Q, Q0 = 1	1001	00000110	00000000
	3. srl Div	1001	00000011	00000000

Problem 3. Convert -4563_{10} into a 32-bit two's complement binary number.

Solution.

$$
\begin{aligned}
4563_{10} &\implies 00000000000000000001000111010011_2 && \text{(Convert to binary, ignore sign)}\\
&\implies 11111111111111111110111000101100_2 && \text{(Invert bits since negative)}\\
&\implies 11111111111111111110111000101101_2 && \text{(Add one)}
\end{aligned}
$$

Problem 4. What decimal number does this two's complement binary number represent: $11111111111111111111001110000011_2$?

Solution.

$$
\begin{aligned}
11111111111111111111001110000011_2 &\implies 00000000000000000000110001111100_2\\
&\qquad\qquad \text{(Invert bits)}\\
&\implies 00000000000000000000110001111101_2\\
&\qquad\qquad \text{(Add one)}\\
&\implies -3197_{10}
\end{aligned}
$$

Problem 5. What would the number 18653.4140625_{10} be in IEEE 754 single precision floating point format?

Solution.

$$
\begin{aligned}
18653_{10} &= 100100011011101_2\\
0.4140625_{10} &= 0.0110101_2\\
18653.4140625_{10} &= 100100011011101.0110101_2 && \text{(Decimal to binary)}\\
&= 1.0010001101110101101 01_2 \times 2^{14} && \text{(Normalized format)}\\
&= 1.001000110111010110101_2 \times 2^{141-127} && \text{(With bias)}\\
141_{10} &= 10001101_2 && \text{(Convert exponent to binary)}
\end{aligned}
$$

$$01000110100100011011101011010100_2 = \text{0x4691BAD4}$$

Problem 6. What would the number -18472.40625_{10} be in IEEE 754 single precision floating point format?

Solution.

$$
\begin{aligned}
18472_{10} &= 100100000101000_2 \\
0.40625_{10} &= 0.01101_2 \\
18472.40625_{10} &= 100100000101000.01101_2 && \text{(Decimal to binary)} \\
&= 1.0010000010100001101_2 \times 2^{14} && \text{(Normalized format)} \\
&= 1.0010000010100001101_2 \times 2^{141-127} && \text{(With bias)} \\
141_{10} &= 10001101_2 && \text{(Convert exponent to binary)}
\end{aligned}
$$

$$11000110100100000101000011010000_2 = \text{0xC69050D0}$$

Problem 7. What decimal number would the IEEE 754 single precision floating point number 0xC5A3B760 (this is in hex) be? Write your final answer in scientific notation as $m \times 10^p$ where p is an integer.

Solution.

$$
\begin{aligned}
0xC5A3B760 &= 11000101101000111011011101100000_2 \\
&\implies -1.01000111011011101100000_2 \times 2^{12} && \text{(Normalized format)} \\
&= -1.279033661_{10} \times 2^{12} \\
&= -5238.921875_{10} \\
&= -5.238921875_{10} \times 10^3
\end{aligned}
$$

Problem 8. For this problem, assume 5 bits precision. Add two binary numbers, $1.0011_2 \times 2^{-8}$ and $1.0101_2 \times 2^{-6}$.

Solution.

$$
\begin{aligned}
1.0011_2 \times 2^{-8} + 1.0101_2 \times 2^{-6} &= 0.010011_2 \times 2^{-6} + 1.0101_2 \times 2^{-6} \\
&= 1.100111_2 \times 2^{-6} && \text{(No overflow/underflow)} \\
&= 1.1001_2 \times 2^{-6} && \text{(Truncated)}
\end{aligned}
$$

Problem 9. For this problem, assume 5 bits precision. Multiply two binary numbers, $1.0011_2 \times 2^{-8}$ and $1.0101_2 \times 2^{-6}$.

$$
\begin{aligned}
1.0011_2 \times 2^{-8} \times 1.0101_2 \times 2^{-6} &= 1.10001111_2 \times 2^{-14} && \text{(No overflow/underflow)} \\
&= 1.1000_2 \times 2^{-14} && \text{(Truncated)}
\end{aligned}
$$

Problem 10. Add $8.96_{10} \times 10^{10}$ to $6.87_{10} \times 10^{8}$, assuming the following two different ways:

(a) You have only three significant digits, first with guard (2 digits) and round digits.

Solution.

$$\begin{aligned} 8.96_{10} \times 10^{10} + 6.87_{10} \times 10^{8} &= 8.96_{10} \times 10^{10} + 0.0687_{10} \times 10^{10} && \text{(Guard digits)} \\ &= 9.0287_{10} \times 10^{10} && \text{(No overflow/underflow)} \\ &= 9.03_{10} \times 2^{10} && \text{(Rounded)} \end{aligned}$$

(b) You have only three significant digits without guard and rounding.

Solution.

$$\begin{aligned} 8.96_{10} \times 10^{10} + 6.87_{10} \times 10^{8} &= 8.96_{10} \times 10^{10} + 0.0687_{10} \times 10^{10} \\ &= 9.0287_{10} \times 10^{10} && \text{(No overflow/underflow)} \\ &= 9.02_{10} \times 2^{10} && \text{(Truncated)} \end{aligned}$$