Lincoln Sand

1) Convert the decimal number -17.6875 into the IEEE 754 format for a single-precision floating-point number. (20 points)

Since the number is negative, the sign bit is 1.

Now we will convert the absolute value (17.6875) into binary.

The integer part is 17, which is 10001 in binary.

The fractional part (0.6875) is converted as follows:

$$0.6875 \cdot 2 = 1.375 \rightarrow 1$$

$$0.375 \cdot 2 = 0.75 \rightarrow 0$$

$$0.75 \cdot 2 = 1.5 \rightarrow 1$$

$$0.5 \cdot 2 = 1.0 \rightarrow 1$$

So, 0.6875 is 0.1011, making the entire number 10001.1011 in binary.

Now we normalize it by shifting the binary point to the right of the first 1 and count the number of shifts. The normalized form is $1.00011011 \cdot 2^4$.

Now we determine the exponent portion. The bias for single precision floating point is 127. Given the actual exponent is 4, the stored component is $4 + 127 = 131$.

In binary, 131 is 10000011.

Now we determine the mantissa. The mantissa is the normalized form excluding the leading 1 and padding the end to 23 bits with 0's if needed.

So, the mantissa is 00011011000000000000000.

Thus, the final representation is:

1 10000011 00011011000000000000000

2) Convert the following IEEE 754 single-precision floating-point register into a decimal number: 0 01111100 01110000000000000000000. (20 points)

The sign bit is 0, so the number is positive.

The exponent is 01111100. In decimal, this is 124. Single precision floating point has a bias of 127, so the actual exponent is $124 - 127 = -3$.

The mantissa is 1.01110000000000000000000.

Converting this to decimal gives:

$$1 + 0 + 0.25 + 0.125 + 0.0625 = 1.4375$$

So, the number is

$$\text{mantissa} \times 2^{\text{exponent}} = 1.4375 \cdot 2^{-3} = 0.1796875$$

The sign bit is positive, so the final number is 0.1796875.

3) Convert the following IEEE 754 double-precision floating-point register into a decimal number: 0 10000000111 1010000000000000000000000000000000000000000000000000. (20 points)

The sign bit is 0, so the final number is positive.

The exponent is 10000000111. In decimal this is 1031. The bias for the exponent for double precision floating point is 1023. So, the actual exponent is $1031 - 1023 = 8$.

The mantissa is:

1.1010000000000000000000000000000000000000000000000000. In decimal, this is:

$$1 + 0.5 + 0 + 0.125 = 1.625$$

The final number is thus:

$$1.625 \cdot 2^8 = 416.0$$

The sign bit is 0, so the final number is 416.0.

4) What is the result of adding the following two IEEE 754 single-precision floating-point registers? Show your steps as you perform the math in binary and produce the final IEEE 754 value that will be placed in the register. (20 points) 0 10000110 01010000000000000000000 0 10000100 11000000000000000000000

Let's first decode the exponents. For the first number, the exponent is 10000110, which is 134 in decimal. For the second number, the exponent is 10000100, which is 132 in decimal.

The bias for single precision floating point is 127, so the actual exponents are $134 - 127 = 7$ and $132 - 127 = 5$ respectively.

We will now align the mantissa's such that the exponents are the same. The first number's mantissa is 1.01010000000000000000000. The second number's mantissa must be shifted right by 2 places, and so it becomes 0.01110000000000000000000.

Now we add these:

$$1.01010000000000000000000 + 0.01110000000000000000000$$
$$= 1.10000000000000000000000$$

No normalization is needed since it is in the interval $[1, 2)$. The larger exponent of the two was 7. We add 1 to this exponent to account for the overflow, resulting in 10000111.

The final number is thus: 0 10000111 10000000000000000000000.

5) Compute the truth table for a logic block that takes in a 3-bit input representing a binary unsigned number Y (values 0 to 7) and produces a 2-bit output X = Y modulo 3. Express each output bit (let's call them X1 and X2) with a sum-of-products Boolean equation. (20 points)

$Y_2Y_1Y_0$ (Decimal)	Y mod 3 (Decimal)	X_1X_0
000 (0)	0	00
001 (1)	1	01
010 (2)	2	10
011 (3)	0	00
100 (4)	1	01
101 (5)	2	10
110 (6)	0	00
111 (7)	1	01

The ouput bit sum-of-products equations for the output bits are:

$$X_1 = \overline{Y_2}Y_1\overline{Y_0} + Y_2\overline{Y_1}Y_0$$

$$X_0 = \overline{Y_2Y_1}Y_0 + Y_2\overline{Y_1Y_0} + Y_2Y_1Y_0$$