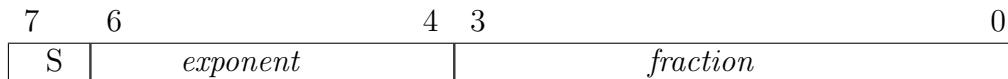


**Question 1:** (20 points)

Assume the same NVIDIA “tiny-precision” floating pointing format from the previous homework, where a floating-point number is represented in 8 bits as follows: the most significant bit is the sign bit, next there are 3 bits used for the exponent, and 4 bits for the fraction. This format is illustrated below:



The exponent is expressed in excess-8 format (also known as a bias representation). Given the binary representation above, the decimal value of the number represented can be computed by the following expression:

$$N = \begin{cases} (-1)^S \times 0.0 & \text{if } exponent = 0 \text{ and } fraction = 0 \\ (-1)^S \times 0.fraction \times 2^{-2} & \text{if } exponent = 0 \text{ and } fraction \neq 0 \\ (-1)^S \times 1.fraction \times 2^{exponent-3} & \text{if } 0 < exponent < 7 \\ (-1)^S \times \infty & \text{if } exponent = 7 \text{ and } fraction = 0 \\ NaN & \text{if } exponent = 7 \text{ and } fraction \neq 0 \end{cases}$$

- a. (5 points) What is the normalized binary representation of  $X = 2.625_{10}$ ? What is the binary representation of  $X$  in the tiny-precision floating-pointing format?
- b. (5 points) What is the normalized binary representation of  $Y = 24_{10}$ ? What is the binary representation of  $Y$  in the tiny-precision floating pointing format?
- c. (5 points) Assume that the floating-point adder in this processor has a guard and a round bit. What is the value of  $X + Y$  computed by this adder? Provide your answer both in **normalized** binary notation and in decimal notation. Show your calculation.
- d. (5 points) Assume that the floating-point adder in a new version of the tiny processor has a guard bit, a round bit and a sticky bit. What is the value of  $X + Y$  computed by this adder? Provide your answer both in **normalized** binary notation and in decimal notation. Show your calculation.