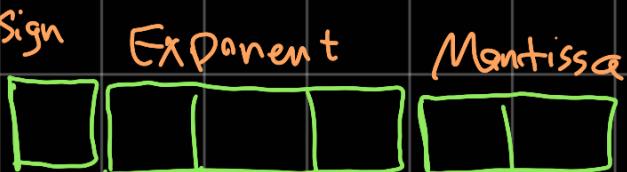


to better understand IEEE 754 standard,  
here I devise my own 6 bit Floating  
Point number!



$$E = (11)_2 = 3 \text{ (half of exponent } (11)_2)$$

$$= (-)^{\text{Sing}} \times a \times 2^{b-E}$$

$\downarrow$   
 a o o } → used to represent non normalized  
 a o 1 } → used to represent normalized  
 o 1 o }  
 o 1 1  
 1 0 0  
 1 0 1 }  
 1 1 0 } → used to represent  $\pm \infty$ , NaN  
 1 1 1 }

★ Normalized and non normalized.

$$(1.a_1a_2)_2 = \left(1 + \frac{1}{2}a_1 + \frac{1}{4}a_2\right) = \text{normalized}$$

$$(0.a_1a_2)_2 = \left(a + \frac{1}{2}a_1 + \frac{1}{4}a_2\right) = \text{non normalized}$$

exponent 000 signals that a number is non normalized.

## \* Possible values for Exponent

6 0 0 → 0  $\xrightarrow{-3}$  -2 (For non normalized)  
 0 0 1 → 1  $\xrightarrow{-3}$  -2  
 0 1 0 → 2  $\xrightarrow{-3}$  -1  
 0 1 1 → 3  $\xrightarrow{-3}$  0  
 1 0 0 → 4  $\xrightarrow{-3}$  1  
 1 0 1 → 5  $\xrightarrow{-3}$  2  
 1 1 0 → 6  $\xrightarrow{-3}$  3  
 1 1 1

\* Possible Values For mantissa

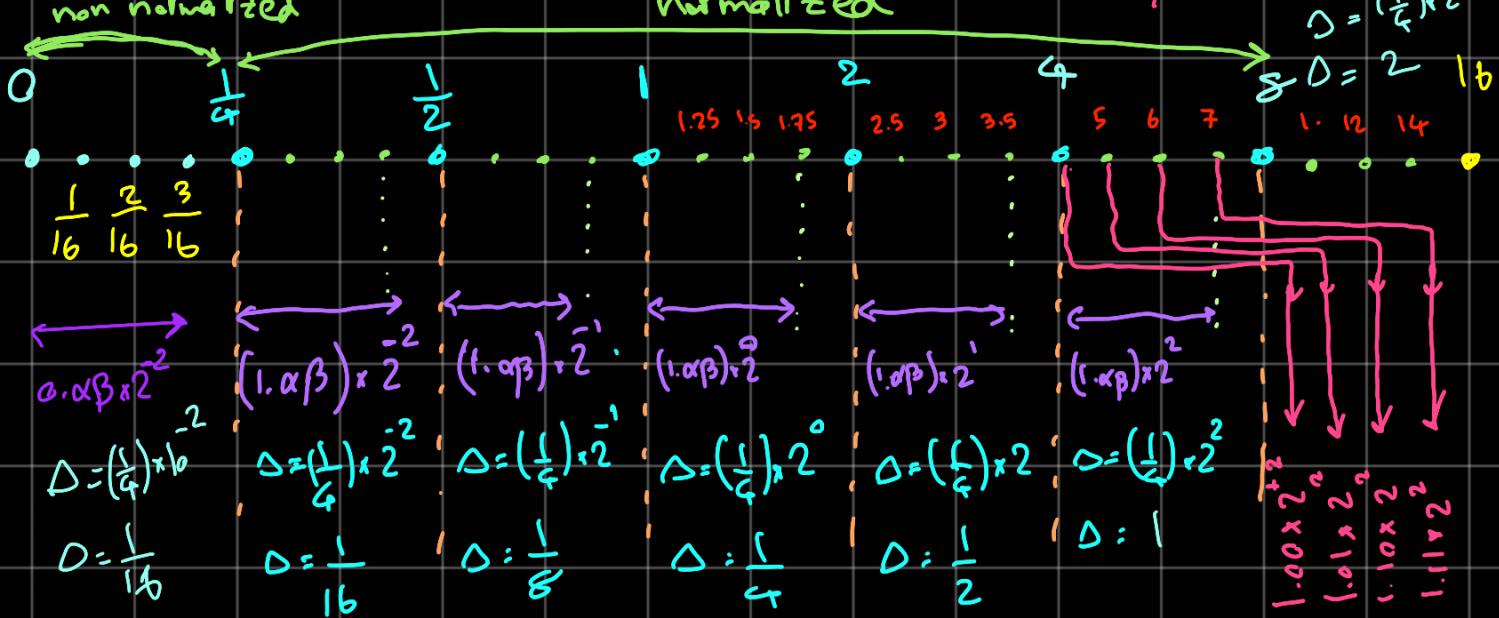
## Normalized

## Non Normalized

$$\begin{matrix} 1.00 \\ 1.01 \\ 1.10 \\ 1.11 \end{matrix} \rightarrow \left( \begin{array}{c} 1+0 \\ 1+\frac{1}{4} \\ 1+\frac{2}{4} \\ 1+\frac{3}{4} \end{array} \right)$$

$$\begin{array}{c} \text{Q.00} \\ \text{Q.01} \\ \text{Q.10} \\ \text{Q.11} \end{array} \xrightarrow{\quad} \left( \begin{array}{c} 0 + 0 \\ 0 + \frac{1}{4} \\ 0 + \frac{2}{4} \\ 0 + \frac{3}{4} \end{array} \right)$$

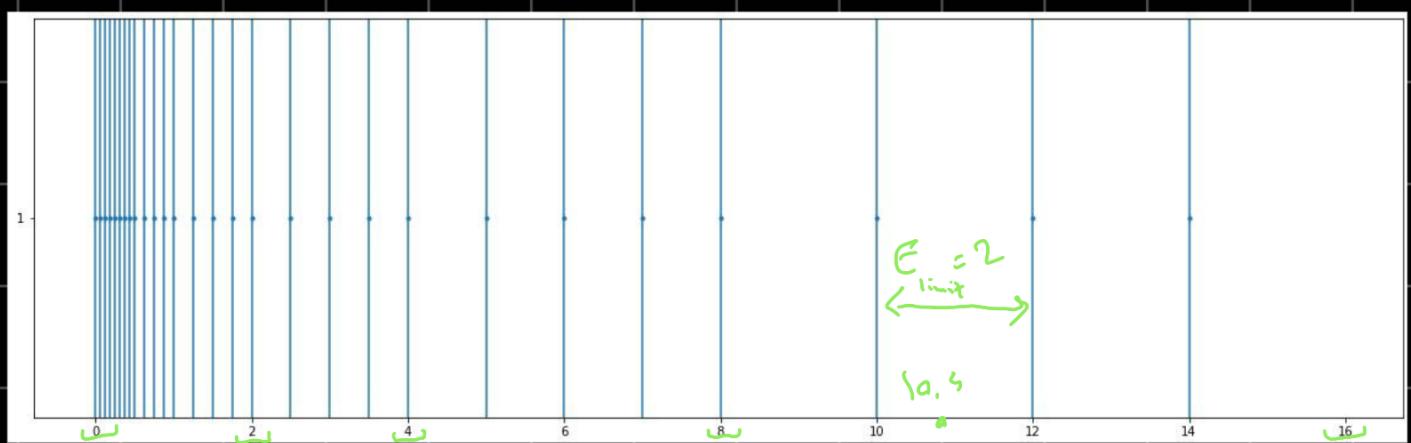
\* Possible values to show with my icee standards  
normalized (1-10?)



So in fact we only have the following numbers in our representation.

|      |        |       |        |                  |
|------|--------|-------|--------|------------------|
| 0    | 0.0625 | 0.125 | 0.1875 | → Non normalized |
| 0.25 | 0.3125 | 0.375 | 0.4375 |                  |
| 0.5  | 0.625  | 0.75  | 0.875  | Normalized       |
| 1    | 1.25   | 1.5   | 1.75   |                  |
| 2    | 2.5    | 3     | 3.5    |                  |
| 4    | 5      | 6     | 7      |                  |
| 8    | 10     | 12    | 14     |                  |

and here is a plot of their spacing



Suppose that our true number is 10.5. If we want to represent it with my representation

We will have  $(1.01) \times 2^3$ . The difference (Error) will be bounded by  $(0.01) \times 2^3 = 2$  (Error will not be bigger than 2)

So the relative error will be:

$$\left( \frac{a - \tilde{a}}{a} \right) \leq \frac{(0.01) \times 2^3}{(1.00) \times 2^3} = (0.01)$$

to calculate the worst error I set  $a$  to be smallest possible

No matter what valid number (in valid range) we want to represent with my IEEE standard, the relative error will always be less than  $(0.01)_2$

which is  $\frac{1}{4}$

Relative error will not be larger than  $\frac{1}{4}$

## Example

True number 13

nearest number in our set  $\Rightarrow$  12

$$\text{relative error} = \frac{13-12}{12} = \frac{1}{12} < \frac{1}{4}$$



True number 5.5

nearest number  $\rightarrow$  5

$$\text{relative error: } \frac{0.5}{5} = \frac{1}{10} < \frac{1}{4}$$

## Example

$$F(x_0 + \Delta h) = F(x_0) + \frac{dF}{dx} \Delta h + \frac{1}{2} \frac{d^2F}{dx^2} \Delta h^2 + \frac{1}{6} \frac{d^3F}{dx^3} \Delta h^3 + \dots$$

$$\stackrel{\text{(est. value)}}{\sim} F(x_0 + \Delta h) = F(x_0) + \frac{dF}{dx} \Delta h$$

$$\text{relative error} = \frac{\stackrel{\sim}{F}(x_0 + \Delta h) - F(x_0 + \Delta h)}{\stackrel{\sim}{F}(x_0 + \Delta h)} = \frac{\frac{1}{2} \frac{d^2F}{dx^2} \Delta h^2 + \dots}{F(x_0 + \Delta h)}$$

we should set the value of  $\Delta h$  in a way  
that the relative error is less than the  
accuracy of our Floating Point number ( $10^{-7}$   
in Case of 32 bit,  $10^{-15}$  For 64 bit and  $\frac{1}{\epsilon}$  For  
my own standard!)

Note that if we use reduced scales  
in our computation/simulation, then  
 $F(x+\Delta x) \approx F(x)$  will not be so big  
or so small. will be around 1.