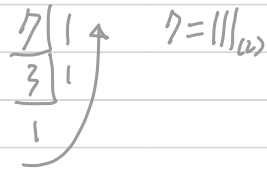
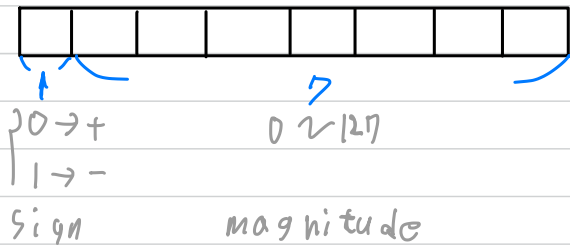


숫자는 수치 정보인가? 문자인가? 이미지인가?
 문자 -> ASCII, Uni Code 사용
 이미지 -> 픽셀로?
 수치정보와 문자정보의 차이점? -> 계산에 목적이 있다.
 인코딩은 "계산"을 할 수 있는 형태로 되어야 한다.

이진수의 변환?
 8bits를 사용한다고 가정하자.
 그러면 $2^8 = 256$ 개가 표현 가능하다.
 100 -> 4



정수 VS 실수? 부호? bit 제한?
 1. bit수 제한?
 2. 부호?
 3. 그렇게 인코딩해서 계산이 효율적인가?



문제는 +0, -0 두가지가 존재한다는 것이다.
 계산은 어떻게 하나? 효율적이지 않다.
 연산에 적합한 것은 2's complement이다.

00001000 : +8
 11110111 : -8

01111100 : +125
 10000011 : -125

2의 보수

1의 보수처럼 0이 두개이고, 계산이 혼란한 것을 막기 위함이다.

양수는 1의 보수와 동일하게 표현하나,

음수는 뒤집어준 뒤, 1을 더해준다. $\rightarrow +127 \sim -128$

--	--	--	--	--	--	--	--

양수는 그냥 만들면 되는데,

음수는 어떻게 만드는가?

양수를 이진법으로 만든 뒤, 1의 보수를 취하고 1을 더해준다.

-124?

124 \rightarrow 01111100

1의 보수 \rightarrow 10000011

2의 보수 \rightarrow 10000100

$$\begin{array}{r} |101| + |011| = 11000 \\ -3 \quad -5 \quad -8 \end{array}$$

$$\begin{array}{r} 1101 \\ 1011 \\ \hline 11000 \end{array}$$

2의 보수로 바꿔주는 회로와, 2진 연산을 하는 회로가 있으면 모든 연산이 가능!

Excess_128 : 표현된 숫자에 128을 뺀다.

$$\begin{array}{rcl} 0 & 105 & 128 \\ -128 & +128 & 255 \\ & -13 & 0 \end{array}$$

부동 소수점?

-63.72 \rightarrow ?

$$= -0.6372 \times 10^2$$

위 3가지 정보만 알면 된다.

$$\begin{array}{r} \text{n bits} \quad \text{m bits} \\ \hline 1001 \quad .8125 \\ 9 \end{array}$$

$$1001$$

$$1101$$

$$\frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{8}$$

어떤 숫자라도 범위내에만 있다면 소숫점 이하의 수를 표현가능한가?

\rightarrow 불가능하다.

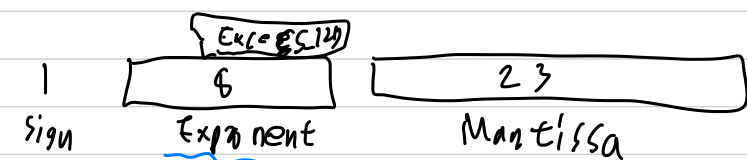
정수부는 만들 수 있다.

당장에 0.2를 만드는 것도 안된다.

0.1은 무한대로 가도 못 만든다.

-111.001101

↓
-1.11001101 × 2³



Excess-127:

부호를 비트 소모 없이 사용가능

$$0.75_{(10)} \rightarrow 0.11_{(2)} = 1.1 \times 2^{-1}$$

$$\frac{0}{+} \quad \frac{1000000}{126(-1)} \quad \frac{1000000000}{mantissa}$$

$$1, \overbrace{01111}^{16 \text{ bits}} | 00, \overbrace{110011}^{16 \text{ bits}}$$

- 124 $\frac{1}{2} \frac{1}{4}$ $\frac{1}{32} \frac{1}{64}$

→ -3

$$\underbrace{1.10011}_{\text{mantissa}} \times 2^{-3}$$

$$\underline{0.001101}$$

8 bits, Excess-127

→ 이진법