NUMBER REPRESENTATION

Why we study number representation

- To meet design metrics minimum number of bits that result in correct answer are required
- Additional bits will results in:
  - Additional Hardware
  - Increases power (Because of more switching of gates and wires)
  - May increase latency

Fixed point integer

- Unsigned (Non-negative integer)
  - For n-bit number, range is given by

$$[0, 2^n - 1]$$

  - Ex. 111
    range [0, 7]

Fixed point integer

- Signed
  - Sign magnitude
    - For n-bit number, range is given by:

      $[-(2^{n-1} - 1), 2^{n-1} - 1]$

    - Ex. $0\,111 \rightarrow 7$ $\quad$ $1\,111 \rightarrow -7$

      Sign bit Magnitude $\quad$ Sign bit Magnitude
  - Signed 2's complement
    - For n-bit number, range is given by:

      $[-2^{n-1}, 2^{n-1} - 1]$

    - Ex. $1\,1\,1\,1$

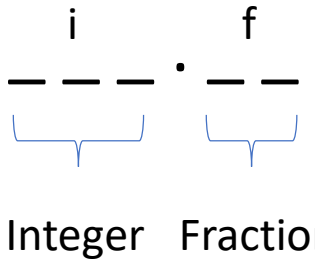
      $-8\ 4\ 2\ 1 \rightarrow -1$

Fixed point integer-Example for different types

Binary number	Unsigned	Sign magnitude	Signed 2's complement
1111	15	-7	-1
1110	14	-6	-2
1000	8	0	-8
0111	7	7	7

Fixed point fraction

- Fraction can be represented as follows



n= i + f

Where n is total no. of bits

- Unsigned range

$[0, 2^{i} - 2^{-f}]$

- Signed (2's complement) range

$[-2^{i-1}, 2^{i-1} - 2^{-f}]$

Sign extension

- Needed for 2’s complement addition
- Consider case of adding two numbers of different widths

```
    1 0 1 1          -5
0 1 0 0 1 0          +18
--------------
0 1 1 1 0 1          +29!
```

Sign extension Rules

- Rule #1: 2’s complement numbers must be the same word-width because of implied zeros

0 0 1 0 1 1	-5
0 1 0 0 1 0	+18
---	
0 1 1 1 0 1	+29! => Still Wrong

Sign extension

- Rule #2: Despite a fundamental change to the number's definition, the value of a 2's complement number will never change due to any amount of sign extension—positive or negative

1 0 1 1 -5= -8 + 2 + 1
1 1 0 1 1 -5= -16 + 8 + 2 + 1
1 1 1 0 1 1 -5= -32 + 16 + 8 + 2 + 1

Sign extension

- Procedure:
  - Calculate the width of the answer word so that it contains all input possibilities
    - It's up to you to make sure the output range is sufficient
  - Extend the inputs' sign bits to the width of the answer
  - Add as usual
  - Ignore bits that ripple to the left of the answer's MSB

1 1 1 0 1 1 -5
0 1 0 0 1 0 +18
---
x 0 0 1 1 0 1 +13