Introduction to Fixed point arithmetic

Digital signal processing is widely used in all our devices.

Comparing to analog signal processing DSP ➔

- more flexibility
- lower power consumption
- higher reliability, higher accuracy
- scalability.

DSP functions can be achieved on different types of hardware processors

- like microcontroller
- ARM
- CPU
- GPU
- SoC
- FPGA

how can we describe a digital signal?

Introduction to Fixed point arithmetic

binary bits to represent a signal.

As you know 1-bit binary contains two states 1 and 0,

which are corresponding to a voltage high and low physically.

The N-bit binary number contains 2^N different stages ➔ 2_N different numbers.

3 bits can represent $2^3 = 8$ different states from 000, 001 to 111.

- **Floating-Point**
- **Fixed-Point**

Introduction to Fixed point arithmetic

The floating-point approach represents and manipulates numbers via N-bit binary in a manner similar to scientific notation.

a number is represented with a mantissa and an exponent

31	30 … 23	22 … 0
SGN	s (8-bit exponent)	m (mantissa)

$$V = (-1)^{SGN} 2^{(s-127)} \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{i} \right)$$

Introduction to Fixed point arithmetic

fixed-point → fixed number of bits N to express fractional numbers

$$2^{(N-M)}\ 2^{M}$$

'M' is the number of bits to express fractional numbers

N = 4

b3	b2	b1	b0	FIX 4.0	FIX 4.1	FIX 4.2	UFIX 4.0	UFIX 4.1	UFIX 4.2
0	0	0	0	0	0.0	0.00	0	0.0	0.00
0	0	0	1	1	0.5	0.25	1	0.5	0.25
0	0	1	0	2	1.0	0.50	2	1.0	0.50
0	0	1	1	3	1.5	0.75	3	1.5	0.75
0	1	0	0	4	2.0	1.00	4	2.0	1.00
0	1	0	1	5	2.5	1.25	5	2.5	1.25
0	1	1	0	6	3.0	1.50	6	3.0	1.50
0	1	1	1	7	3.5	1.75	7	3.5	1.75
1	0	0	0	-8	-4.0	-2.00	8	4.0	2.00
1	0	0	1	-7	-3.5	-1.75	9	4.5	2.25
1	0	1	0	-6	-3.0	-1.50	10	5.0	2.50
1	0	1	1	-5	-2.5	-1.25	11	5.5	2.75
1	1	0	0	-4	-2.0	-1.00	12	6.0	3.00
1	1	0	1	-3	-1.5	-0.75	13	6.5	3.25
1	1	1	0	-2	-1.0	-0.50	14	7.0	3.50
1	1	1	1	-1	-0.5	-0.25	15	7.5	3.75

Introduction to Fixed point arithmetic

How can we handle the negative number?

use the MSB to represent the sign bit ➔ MSB = '1' negative, ‘0’ positive.

b3	b2	b1	b0	1'Compl
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	-0
1	1	1	0	-1
1	1	0	1	-2
1	1	0	0	-3
1	0	1	1	-4
1	0	1	0	-5
1	0	0	1	-6
1	0	0	0	-7

two zeros values:

- positive zero
- negative zero

Introduction to Fixed point arithmetic

2’s complement representation

b3	b2	b1	b0	2'Compl
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	-1
1	1	1	0	-2
1	1	0	1	-3
1	1	0	0	-4
1	0	1	1	-5
1	0	1	0	-6
1	0	0	1	-7
1	0	0	0	-8

Introduction to Fixed point arithmetic

b3	b2	b1	b0	1'Compl
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	-0
1	1	1	0	-1
1	1	0	1	-2
1	1	0	0	-3
1	0	1	1	-4
1	0	1	0	-5
1	0	0	1	-6
1	0	0	0	-7

$x_{1's} = 2^N - 1 - |x|$

$x_{2's} = x_{1's} + 1 = \mathbf{2^N - |x|}$

$-6 = 2^4 - 10$

10d = 1010b

b3	b2	b1	b0	2'Compl
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	-1
1	1	1	0	-2
1	1	0	1	-3
1	1	0	0	-4
1	0	1	1	-5
1	0	1	0	-6
1	0	0	1	-7
1	0	0	0	-8

Introduction to Fixed point arithmetic

Example: 5-2

the operation can be written as 5 + (-2)

If we are using 4 bit, the 2'compl representation of -2 is:

$-2 = 2^4 - 2 = 16-2=14$

so $(5 - 2)_{2'c} = 5+14 =$

```
 0101+
 1110
=======
 0011b = 3d
```