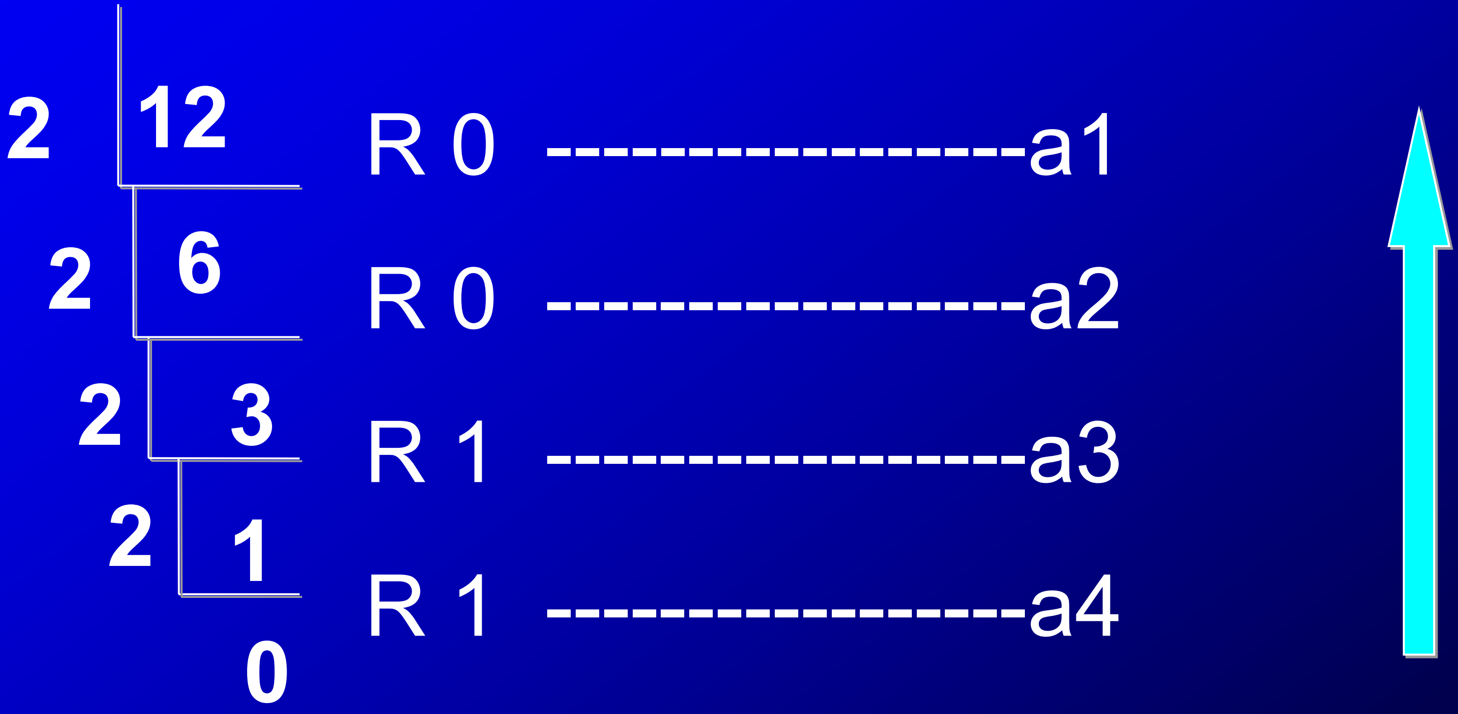
Preparation: Conversion between decimal and binary

- What is the result of

0.75 - 0.5 - 0.25?

0.6 - 0.3 - 0.2 - 0.1?

1、Convert $(12)_{10}$ to binary



2 12 R 0 -----------------a1

2 6 R 0 -----------------a2

2 3 R 1 -----------------a3

2 1 R 1 -----------------a4

0

$\therefore (12)_{10}=(1100)_2$

2、Convert $(0.25)_{10}$ to a binary number

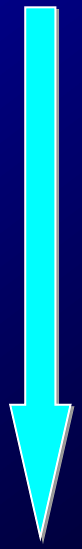
```
  0.25
 ×  2
---------
  0.5            | 0------a1
 ×  2
---------
    1            | 1------a2
```

$\therefore (0.25)_{10}=(0.01)_2$

3、Convert $(0.6)_{10}$ to a binary number

```
  0.6
 × 2
---------
  0.2              | 1------a1
 × 2
---------
  0.4              | 0------a2

          ⋮
```



https://tool.oschina.net/hexconvert/

4、Convert $(7/32)_{10}$ to a binary number

1/32 = 0.00001

7 = 111

7 * (1/32) = 0.00111

Chapter2 Numerical representation and calculation

2.1 Unsigned number and signed number

2.2 Fixed point and floating point representation

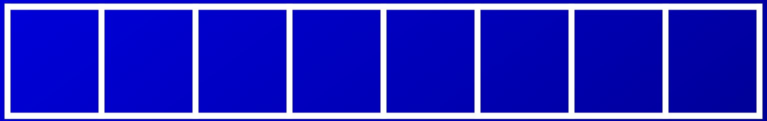
2.3 Arithmetic and Logical Operation Basis

2.4 Fixed point operations

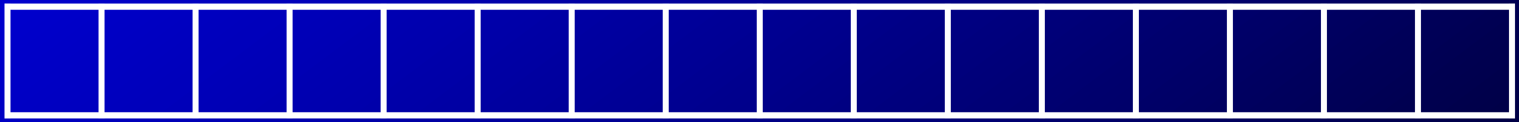
2.1 Unsigned number and signed number

1、 Unsigned number

The **number of bits in the register** reflects the **range** of representation of the **unsigned number.**



8 bits 0 ~ 255

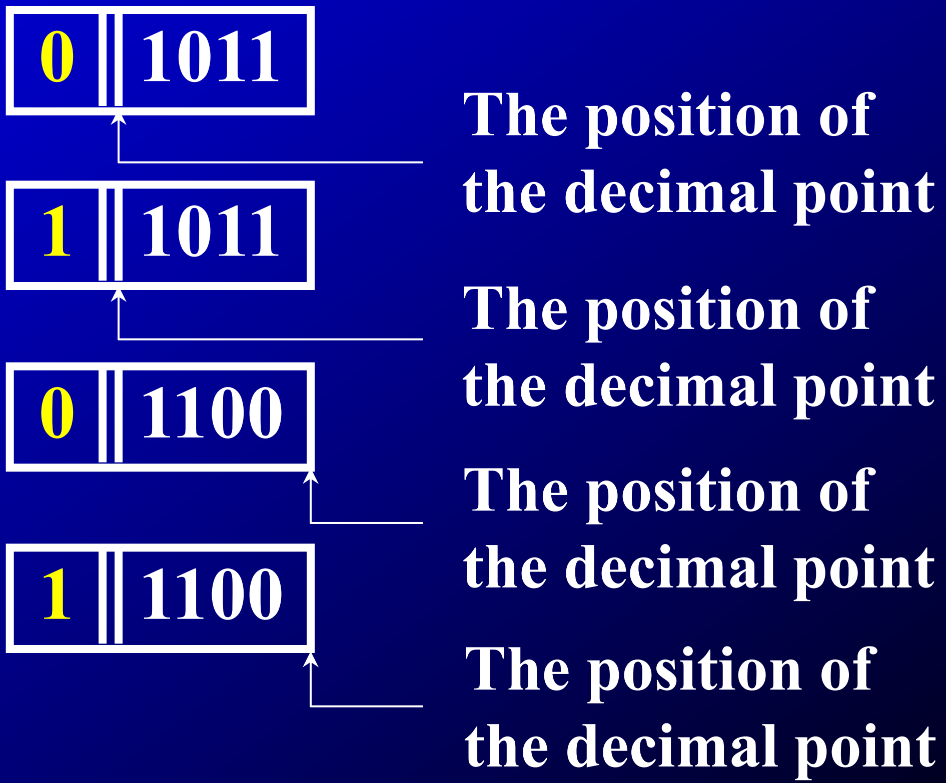


16 bits 0 ~ 65535

2、Signed number

(1)Machine number(机器数) and true value(真值)

True value	Machine number
Signed number	Symbolic digitized number
+ 0.1011	0 1011
– 0.1011	1 1011
+ 1100	0 1100
– 1100	1 1100



The position of the decimal point

The position of the decimal point

The position of the decimal point

The position of the decimal point

(2)True form(原码) representation

(a) Definition

Integer (整数)

$$[x]_{\mathrm{TF}} = \begin{cases} 0,\ \ x & 2^n > x \geq 0 \\ 2^n - x & 0 \geq x > -2^n \end{cases}$$

x is the true value n is the number of bits

Example: $x = +1110$ $[x]_{\mathrm{TF}} = 0\,,\,1110$

Separate the sign bit from the numeric part with a ‘,’

$x = -1110$ $[x]_{\mathrm{TF}} = 2^4 + 1110 = 1\,,\,1110$

Decimal part (小数)

$$[x]_{\mathrm{TF}} = \begin{cases} x & 1 > x \geq 0 \\ 1 - x & 0 \geq x > -1 \end{cases}$$

x is True value

Example:

$x = +0.1101 \qquad [x]_{\mathrm{TF}} = 0\,.\,1101$

Use a '.' to separate the sign bit from the numeric part

$x = -0.1101 \qquad [x]_{\mathrm{TF}} = 1 - (-0.1101) = 1\,.\,1101$

$x = +0.1000000 \qquad [x]_{\mathrm{TF}} = 0\,.\,1000000$

Use a '.' to separate the sign bit from the numeric part

$x = -0.1000000 \qquad [x]_{\mathrm{TF}} = 1 - (-0.1000000) = 1\,.\,1000000$

(b) Example

Example 2.1 Given $[x]_{TF} = 1.\underline{0011}$ find x – 0.0011

Solve: from the definition

$$x = 1 - [x]_{TF} = 1 - 1.0011 = - 0.0011$$

Example 2.2 Given $[x]_{TF} = 1,\underline{1100}$ find x – 1100

Solve: from the definition

$$x = 2^4 - [x]_{TF} = 10000 - 1,1100 = - 1100$$

Example 2.3 Given $[x]_{\text{TF}} = 0.1101$ find x

Solve: $\because [x]_{\text{TF}} = 0.1101$

$\therefore \quad x = +\,0.1101$

Example 2.4 Find the true form of $x = 0$ (n=4)

Solve:

Suppose $x = +0.0000$ $\quad [+0.0000]_{\text{TF}} = 0.0000$

$x = -\,0.0000$ $\quad [-0.0000]_{\text{TF}} = 1.0000$

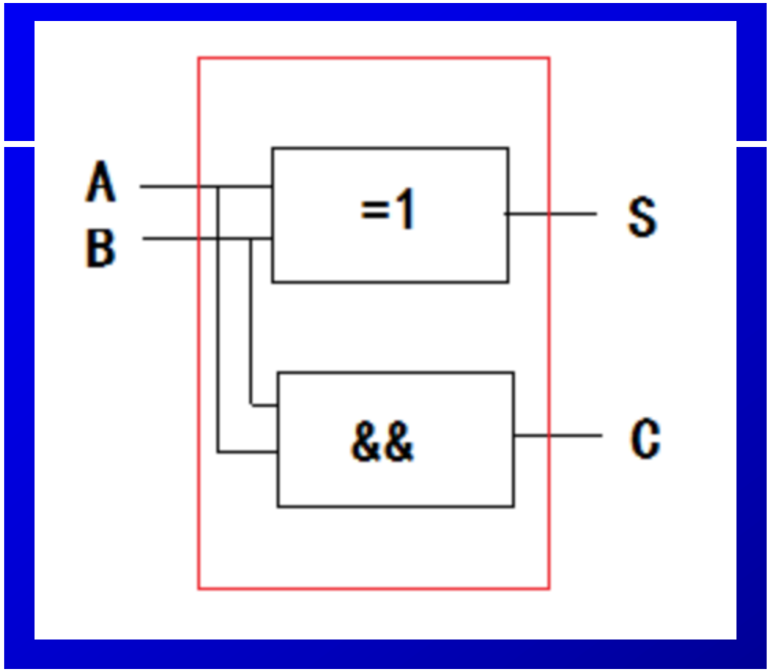
Similarly, for Integer $[+\,0]_{\text{TF}} = 0{,}0000$

$[-\,0]_{\text{TF}} = 1{,}0000$

$\therefore \quad [+0]_{\text{TF}} \neq [\;-0]_{\text{TF}}$

True form features: Simple and intuitive

However, when doing + operation with the True form, the following problem occurs:



Operation	Result
Add	+
Sub	+/-
Sub	+/-
Add	-

Can **subtraction** be operated by **addition operation** ?

https://www.dsvlab.cn/vlab/vlab.html

(3) Two's Complement （补码）

(a) Concept of complement (补的概念)

- **Clock** anti-clockwise

$$\begin{array}{r} 6 \\ -\ 3 \\ \hline 3 \end{array}$$

Clockwise

$$\begin{array}{r} 6 \\ +\ 9 \\ \hline 15 \\ -\ 12 \\ \hline 3 \end{array}$$

− 3 can be replaced by + 9 **Sub ⟶ Add**

We call + 9 is the **complement** of -3 modulo 12

mod(15, 12)

Denoted by– 3 ≡ + 9 （mod 12）

– 4 ≡ + 8 （mod 12）

− 5 ≡ + 7 （mod 12）

Conclusion

□ The complement of a negative number is obtained by adding "module(模)"

□ The complement of a positive number is itself

- （mod 16）　$1011 \longrightarrow 0000$?

$$\begin{array}{r} 1011 \\ -\ 1011 \\ \hline 0000 \end{array} \qquad\qquad \begin{array}{r} 1011 \\ +\ 0101 \\ \hline \boxed{1}0000 \end{array}$$

eliminate

We can replace -1011 by + 0101

Denoted by $-1011 \equiv +0101$ （mod 2^4）

Similarly $-011 \equiv +101$ （mod 2^3）

Examples

$$-3 \equiv +7 \pmod{10}$$

$$+7 \equiv +7 \pmod{10}$$

$$-3 \equiv +97 \pmod{10^2}$$

$$+97 \equiv +97 \pmod{10^2}$$

$$-1101 \equiv +0011 \pmod{2^4}$$

Examples

- **Suppose A = 9, B = 5, solve A-B (mod 32)**
  - A–B = 9 – 5 = 4

- – 5 ≡ +27 (mod 32)
- Then A – B = 9+27=36 (add)
- 4 = 36 mod 32

- **What if 5 – 9 (mod 16)?**

(b) Definition of 2’s Complement

Integer（整数）

$$[x]_{2'} = \begin{cases} 0,\ x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \pmod{2^{n+1}} \end{cases}$$

x is the true value　　n is the number of bits

For　　$x = +1010$　　　　$x = -1011000$

$[x]_{2'} = 0,1010$　　　　$[x]_{2'} = 2^{7+1} + (-1011000)$

$$\begin{array}{rr} = & 100000000 \\ - & 1011000 \\ \hline & 1,0101000 \end{array}$$

Separate the sign bit from the numeric part with a ‘,’

$x = -1010$?

Decimal part (小数)

$$[x]_{2'} = \begin{cases} x & 1 > x \geq 0 \\ 2 + x & 0 > x \geq -1 \quad (\text{mod } 2) \end{cases}$$

x is the true value

Example $x = +\,0.1110$ $\qquad\qquad x = -\,0.1100000$

$[x]_{2'} = 0.1110$

$$\begin{aligned} [x]_{2'} &= 2 + (\,-0.1100000\,) \\ &= 10.0000000 \\ &-\ \ 0.1100000 \\ \hline &\ \ \ 1.0100000 \end{aligned}$$

Use a **decimal point** to separate the sign bit from the numeric part

(c) Shortcut for 2’s complement

Suppose $x = -1010$,

Then $[x]_{2'} = 2^{4+1} - 1010 = 11111 + 1 - 1010$

$$\begin{array}{r} = 100000 \\ -\ 1010 \\ \hline = \ 1,0110 \end{array} \qquad \begin{array}{r} = 11111\ \ +1 \\ -\ 1010\qquad \\ \hline \boxed{10101} + 1 \\ = 1,0110\qquad \end{array}$$

and $[x]_{TF} = \boxed{1,1010}$

When the true value is negative, the 2’s complement can be obtained by taking the invert of the True Form except the symbol bit, then adding 1 to the least significant bit（补码 可用 原码除符号位外按位取反，末位加一获得）

(d) Examples

Example 2.4 Known $[x]_{2'} = 0.0001$

Find x

Solve：From the definition: $x = +\,0.0001$

Example 2.5 Known $[x]_{2'} = 1.0001$

Find x

Solve：From the definition:

$$x = [x]_{2'} - 2$$

$$= 1.0001 - 10.0000$$

$$= -\,0.1111$$

$$[x]_{2'} \xrightarrow{?} [x]_{TF}$$

$$[x]_{TF} = 1.1111$$

$$x = -0.1111$$

(d) Examples

Example 2.6 Known $[x]_{2'} = 1,1110$

Solve x

Solve: From the definition: $[x]_{2'} \xrightarrow{?} [x]_{TF}$

$x = [x]_{2'} - 2^{4+1}$ $\quad [x]_{TF} = 1,0010$

$= 1,1110 - 100000$ $\quad \therefore x = -0010$

$= -0010$

When the 2’s complement is negative, the true form can be obtained by taking the invert of the 2’ complement except the symbol bit, then adding 1 to the least significant bit

Exercise Find 2’s complement of the true values

True value	$[x]_{2'}$	$[x]_{TF}$
$x = +70 \;=\; +1000110$	0, 1000110	0,1000110
$x = -70 \;=\; -1000110$	1, 0111010	1,1000110
$x = +0.1110$	0.1110	0.1110
$x = -0.1110$	1.0010	1.1110
$x = \boxed{+0.0000}$ $[+0]_{2'} = [-0]_{2'}$	$\boxed{0.0000}$	0.0000
$x = \boxed{-0.0000}$	$\boxed{0.0000}$	1.0000
$x = -1.0000$	1.0000	NULL

Defined by 2’s complement of decimal :

$$[x]_{2'} = \begin{cases} x & 1 > x \geq 0 \\ 2 + x & 0 > x \geq -1 \ (\mathrm{mod}\ 2) \end{cases}$$

$$[-1]_{2'} = 2 + x = 10.0000 - 1.0000 = 1.0000$$

Convert subtraction into addition

- 0011-0110：

$$\begin{array}{r} 0{,}0011 \\ +\ 1{,}1010 \\ \hline 1{,}1101 \end{array}$$

- 0110-0011：

$$\begin{array}{r} 0{,}0110 \\ +\ 1{,}1101 \\ \hline 10{,}0011 \end{array}$$

（4）One’s-complement (反码)

(a) Definition

Integer

$$[x]_{1'} = \begin{cases} 0,\ x & 2^n > x \geq 0 \\ (2^{n+1} - 1) + x & 0 \geq x > -2^n \ (\text{mod } 2^{n+1} - 1) \end{cases}$$

x is the true value　　n is the number of bits

Such as $x = +1101$　　　　$x = -1101$

$$[x]_{1'} = 0{,}1101$$

$$[x]_{1'} = (2^{4+1} - 1) - 1101$$
$$= 11111 - 1101$$
$$= 1{,}0010$$

Separate the sign bit
from the numeric
part with a ‘,’

Decimal part (小数)

$$[x]_{1'} = \begin{cases} x & 1 > x \geq 0 \\ (2 - 2^{-n}) + x & 0 \geq x > -1 \ (\mathrm{mod}\ 2 - 2^{-n}) \end{cases}$$

x is the true value *n* is the number of bits

Examples:

$$x = +0.1101$$

$$[x]_{1'} = 0.1101$$

$$x = -0.1010$$

$$[x]_{1'} = (2 - 2^{-4}) - 0.1010$$

$$= 1.1111 - 0.1010$$

$$= 1.0101$$

Separate the sign bit
from the numeric
part with a ‘.’

(b) Examples

Example 2.7 Known $[x]_{1'} = 0{,}1110$ Find x

Solve: From def: $x = +1110$

Example 2.8 Known $[x]_{1'} = 1{,}1110$ Find x

Solve: From def: $x = [x]_{1'} - (2^{4+1} - 1)$

$= 1{,}1110 - 11111$

$= -0001$

Example 2.9 Given the One's-complement of 0

Solve: Sup $x = +0.0000$ $[+0.0000]_{1'} = 0.0000$

$x = -0.0000$ $[-0.0000]_{1'} = 1.1111$

Similarly, for Integer $[+0]_{1'} = 0{,}0000$ $[-0]_{1'} = 1{,}1111$

$\therefore$ $[+0]_{1'} \neq [\quad \theta]_{1'}$

Summary of three machine numbers

□ The highest bit is the sign bit . “,” （Integer） “.” （ Decimals ） to separate them.

□ For positives， True form = 2’ = 1’

□ For negatives， Sign bit is 1, For numerical part

the 2’ complement can be obtained by taking the invert of the original True from except the symbol bit, then adding 1 to the least significant bit

the 1’ complement can be obtained by taking the invert of the original True form except the symbol bit

Suppose the True value of x= -1010， Then, the one’s complement of x should be__.

A. 1,1010 B. 1,0101 C. 0,0101 D. 1,0110

Suppose the true value of x= -0.0101， then the 2’s complement of x is __.

A. 0.0101 B. 1.0101 C. 1.1010 D. 1.1011

Example 2.10 Let the machine number be 8 bits long (1 bit is the sign bit). For integers, when they represented with the form of binary code, True form, 2’s comp and 1’s comp, what is the corresponding range of true value?

Binary Code	True value of unsigned	True value of True form	True value of 2’s comp	True value of 1’s comp
00000000	0	+0	±0	+0
00000001	1	+1	+1	+1
00000010	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮
01111111	127	+127	+127	+127
10000000	128	-0	-128	-127
10000001	129	-1	-127	-126
⋮	⋮	⋮	⋮	⋮
11111101	253	-125	-3	-2
11111110	254	-126	-2	-1
11111111	255	-127	-1	-0

Example 2.11 Given $[y]_{2'}$, find $[-y]_{2'}$

Suppose $[y]_{2'} = y_0.y_1y_2\ldots y_n$

① If $\boxed{[y]_{2'} = 0.y_1y_2\ldots y_n}$

$$-y = -0.y_1y_2\ldots y_n$$

$$\boxed{[-y]_{2'} = 1.\overline{y_1}\,\overline{y_2}\ldots\overline{y_n} + 2^{-n}}$$

② If $\boxed{[y]_{2'} = 1.y_1y_2\ldots y_n}$

$$y = -0.\overline{y_1}\,\overline{y_2}\ldots\overline{y_n} + 2^{-n}$$

$$-y = +0.\overline{y_1}\,\overline{y_2}\ldots\overline{y_n} + 2^{-n}$$

$$\boxed{[-y]_{2'} = 0.\overline{y_1}\,\overline{y_2}\ldots\overline{y_n} + 2^{-n}}$$

Invert the given number and add 1 to the least significant bit(连同符号位按位取反末位加一)

C语言代码如下。

```c
int i=32777;
short si=i;
int j=si;
```

执行上述代码后，j 的值是

A. -32777

B. -32759

C. 32759

D. 32777

（5） Biased Representation (移码)

2’s Complement is difficult to judge its true value directly

Example	Decimal R	Binary R	Two’s Cmp R	
	$x = +21$	$+10101$	0,10101	✗
	$x = -21$	-10101	1,01011	larger
	$x = +31$	$+11111$	0,11111	✗
	$x = -31$	-11111	1,00001	larger

$x + \boxed{2^5}$

$+10101 + 100000 = 110101$ larger

$-10101 + 100000 = 001011$ ✓

$+11111 + 100000 = 111111$ larger

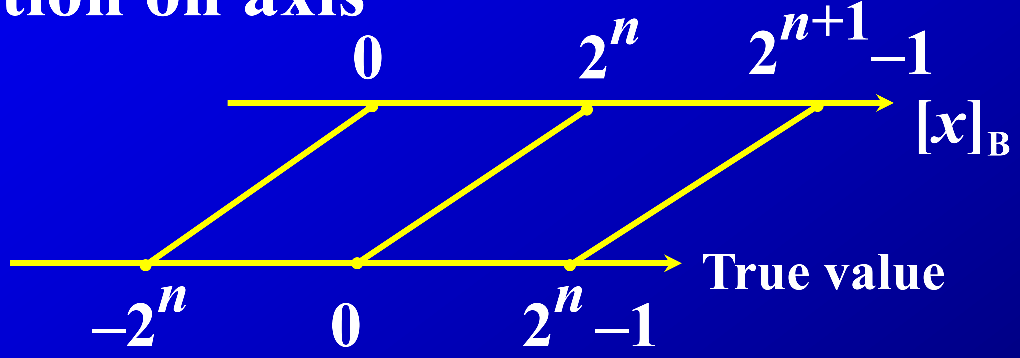
$-11111 + 100000 = 000001$ ✓

(a) Definition

$$[x]_B = 2^n + x \quad (2^n > x \geq -2^n)$$

x is the true value， *n* is the number of bit

Representation on axis



Example: $x = +10100$

$$[x]_B = 2^5 + 10100 = 1{,}10100$$

$$x = -10100$$

$$[x]_B = 2^5 - 10100 = 0{,}01100$$

Separate the sign bit from the numeric part with a ‘,’

(b) Comparison of Biased and 2's Representation

Suppose: $x = +1100100$

$[x]_B = 2^7 + 1100100 = 1,1100100$

$[x]_{2'} = 0,1100100$

Suppose: $x = -1100100$

$[x]_B = 2^7 - 1100100 = 0,0011100$

$[x]_{2'} = 1,0011100$

Only sign bit is different between 2's complement and biased representation

Suppose x=-1010，then the Biased Representation of x is__.

A. 0,0110 **B.** 1,0101 **C.** 1,0110 **D.** 0,0101

(c) True value, 2’s and Biased

True value x (n=5)	$[x]_{2'}$	$[x]_B$	Decimal integer of $[x]_B$
- 1 0 0 0 0 0	1 0 0 0 0 0	0 0 0 0 0 0	0
- 1 1 1 1 1	1 0 0 0 0 1	0 0 0 0 0 1	1
- 1 1 1 1 0	1 0 0 0 1 0	0 0 0 0 1 0	2
⋮	⋮	⋮	⋮
- 0 0 0 0 1	1 1 1 1 1 1	0 1 1 1 1 1	31
± 0 0 0 0 0	0 0 0 0 0 0	1 0 0 0 0 0	32
+ 0 0 0 0 1	0 0 0 0 0 1	1 0 0 0 0 1	33
+ 0 0 0 1 0	0 0 0 0 1 0	1 0 0 0 1 0	34
⋮	⋮	⋮	⋮
+ 1 1 1 1 0	0 1 1 1 1 0	1 1 1 1 1 0	62
+ 1 1 1 1 1	0 1 1 1 1 1	1 1 1 1 1 1	63

(d) Characteristics of Biased R

□ When $x = 0$ $[+0]_B = 2^5 + 0 = 1,00000$

$[-0]_B = 2^5 - 0 = 1,00000$

$\therefore [+0]_B = [-0]_B$

□ When $n = 5$ The Mini true value: $-2^5 = -100000$

$[-100000]_B = 2^5 - 100000 = 000000$

It can be seen that the minimum true value of the Biased Representation is all 0

1.For____，____， the representation of zero is unique.

2.For machine number 80H：

if the true value is ± 0， then it should be ();

if it represents -128， then it should be ();

if it represents -127， then it should be ();

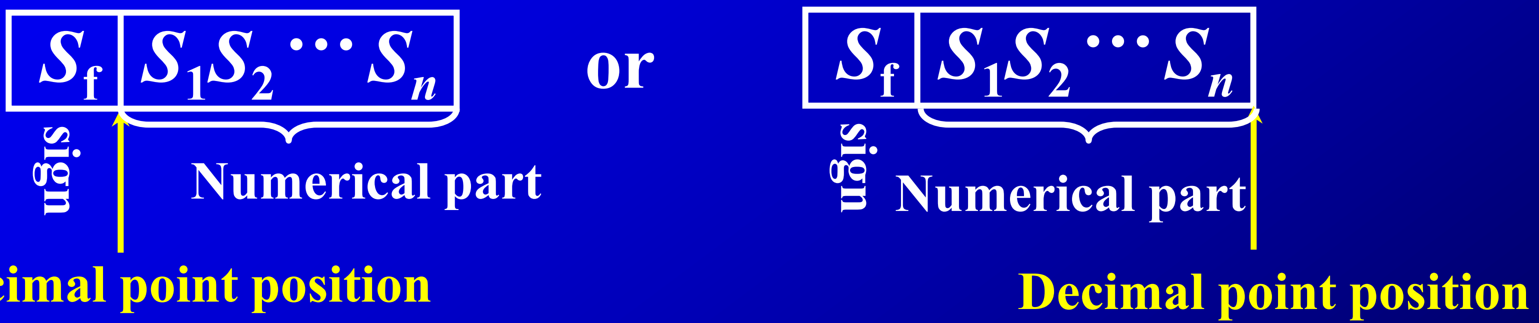
if it represents -0, then it should be ().

A.True form B. One’s comp C.Two’s comp D. Biased

2.2 Fixed point and floating point representation

The decimal point is marked by default

1. Fixed point representation



$\boxed{S_f}\boxed{S_1S_2\cdots S_n}$ or $\boxed{S_f}\boxed{S_1S_2\cdots S_n}$

sign

Numerical part

Decimal point position

sign

Numerical part

Decimal point position

Fixed	Decimal(小数)	Integer(整数)
TF	$-(1-2^{-n}) \sim +(1-2^{-n})$	$-(2^n-1) \sim +(2^n-1)$
2’s	$-1 \sim +(1-2^{-n})$	$-2^n \sim +(2^n-1)$
1’s	$-(1-2^{-n}) \sim +(1-2^{-n})$	$-(2^n-1) \sim +(2^n-1)$

2. Floating number representation

Floating number: are numbers that contain floating decimal points.

$N = S \times b^{E}$ General form of floating point numbers

S 尾数Significand E 阶码Exponent b 基数base

Mantissa

binary

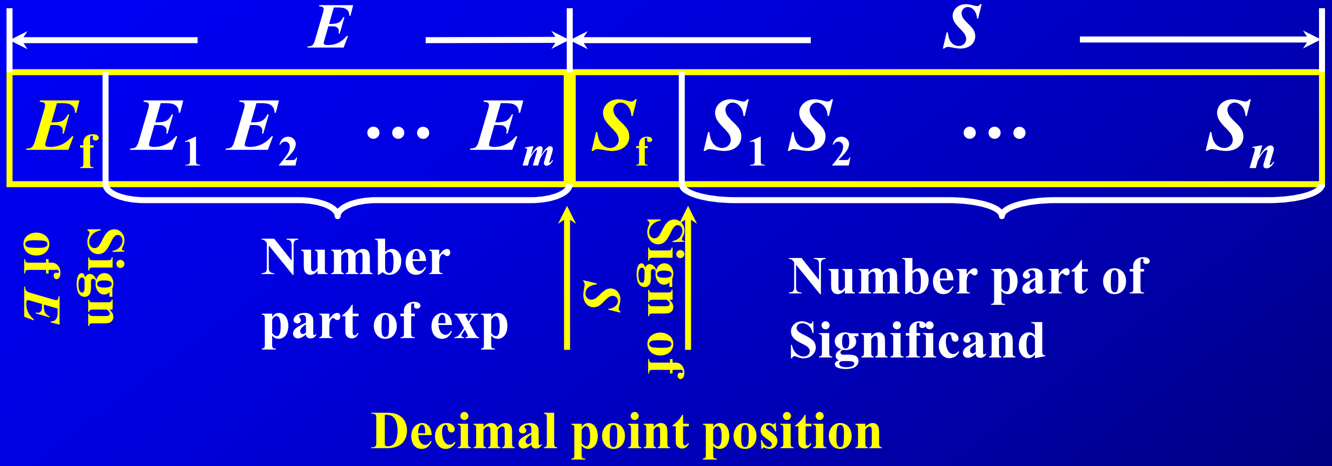
when $b = 2$ $N = 11.0101$

$$
\begin{aligned}
&\checkmark = 0.110101 \times 2^{\boxed{10}} \quad \text{normalized form (规格化)} \\
&= 1.10101 \times 2^{1} \\
&= 1101.01 \times 2^{-10} \\
&= 0.00110101 \times 2^{100}
\end{aligned}
$$

S Decimal, can be positive or negative

E Integer, can be positive and negative

(a). Floating number representation


E
S
E_f E_1 E_2 … E_m
S_f S_1 S_2 … S_n
Sign of E
Number part of exp
Sign of S
Number part of Significand
Decimal point position

S_f Sign of floating number(浮点数符号)

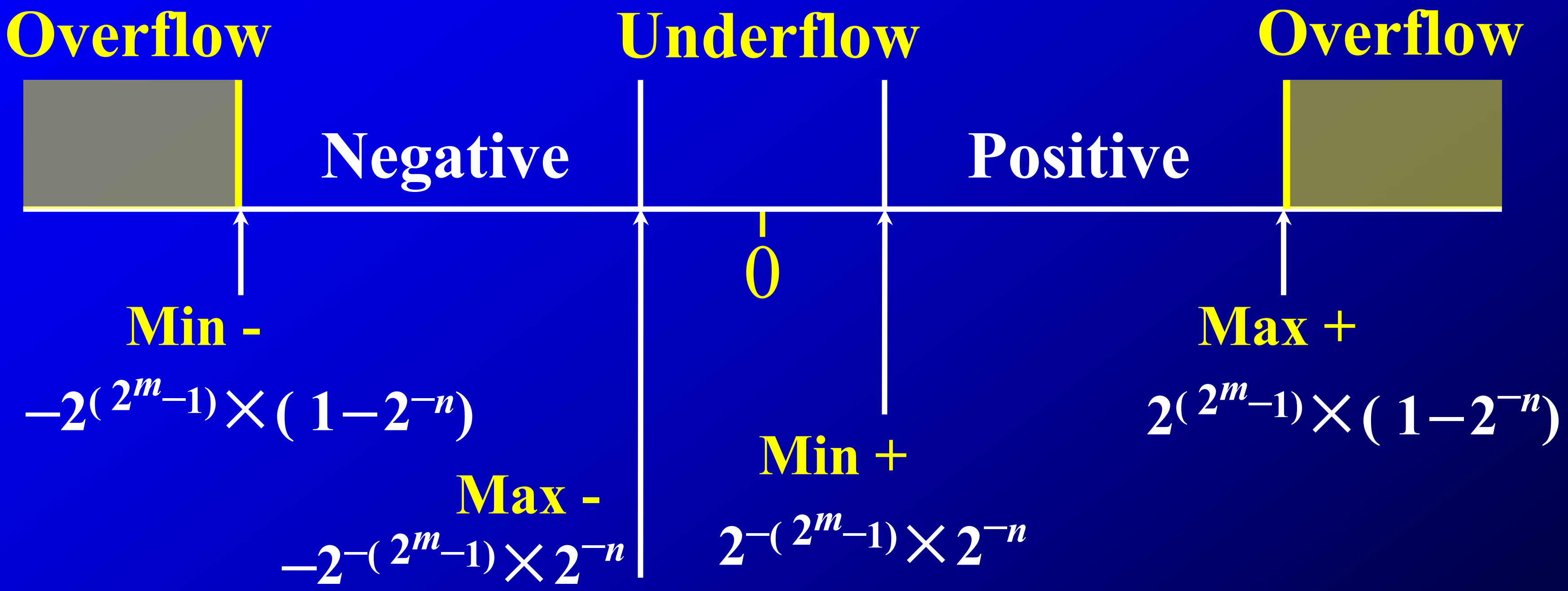
n Number of bits reflects the precision(精度)

m The range of floating number(表示范围)

(b). Representation range

Overflow: E > maximum E Error

Underflow: E < minimum E machine zero

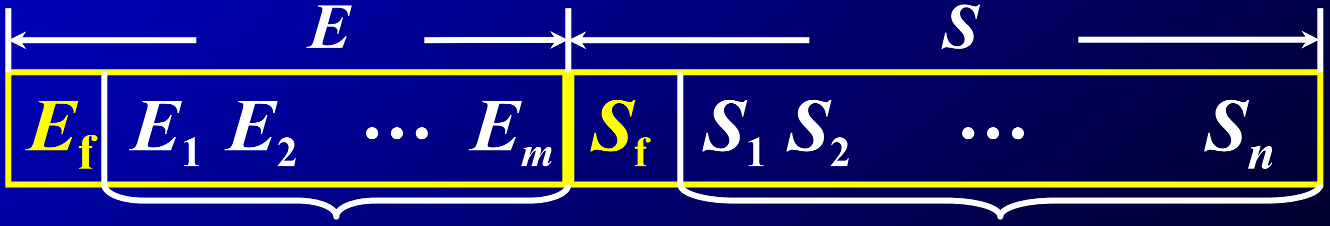


Min -: $-2^{(2^m-1)}\times(1-2^{-n})$

Max +: $2^{(2^m-1)}\times(1-2^{-n})$

Min +: $2^{-(2^m-1)}\times 2^{-n}$

Max -: $-2^{-(2^m-1)}\times 2^{-n}$



(c). Exercises

Let the machine number be 24 bits long, we want to represent ± 30,000 decimal numbers. In the premise of guaranteeing the maximum precision, what is the bits of *E* and *S* , except for the sign of *E* and the Sign of *S*;

设机器数字长为 24 位，欲表示±3万的十进制数，试问在保证数的最大精度的前提下，除阶符、数符各 取1 位外，阶码、尾数各取几位？

Solve： ∵ $2^{14} = 16384$ $2^{15} = 32768$

∴ 2^15 can represent ±30000 decimal numbers

$$2^{\boxed{15}} \times 0.\underbrace{\times\times\times \; \cdots \; \times\times\times}_{n}$$

$m = 4(2^4\text{-}1=15)$， $5, \ldots$

Then $m = 4$， $n = 18$

(c). Exercises

- Suppose the floating number with the length of 16 bits, inside which *E*=5 bits (contains 1 bit for sign), *S*=11bits(1 bit for sign), to convert the decimal 13/128 into binary type with fixed-point number and floating number representation respectively.

Solve：

Binary type: $x = 0.0001101$

Fixed-point: $x = 0.0001101000$

Floating norm: $x = 0.1101000000 \times 2^{-0011}$

Fixed point machine $[x]_{TF} = [x]_{2'} = [x]_{1'} = 0.\ 0001101000$

Floating point machine $[x]_{TF} = 1,\ 0011;\ \ 0.\ 1101000000$

$[x]_{2'} = 1,\ 1101;\ \ 0.\ 1101000000$

$[x]_{1'} = 1,\ 1100;\ \ 0.\ 1101000000$

(c). Exercises

Use binary fixed-point and floating point number to represent -58, and provide its representations in fixed and floating machines. (Other requirements are identical)

Solve: Let $x = –58$

Binary type: $x = – 111010$

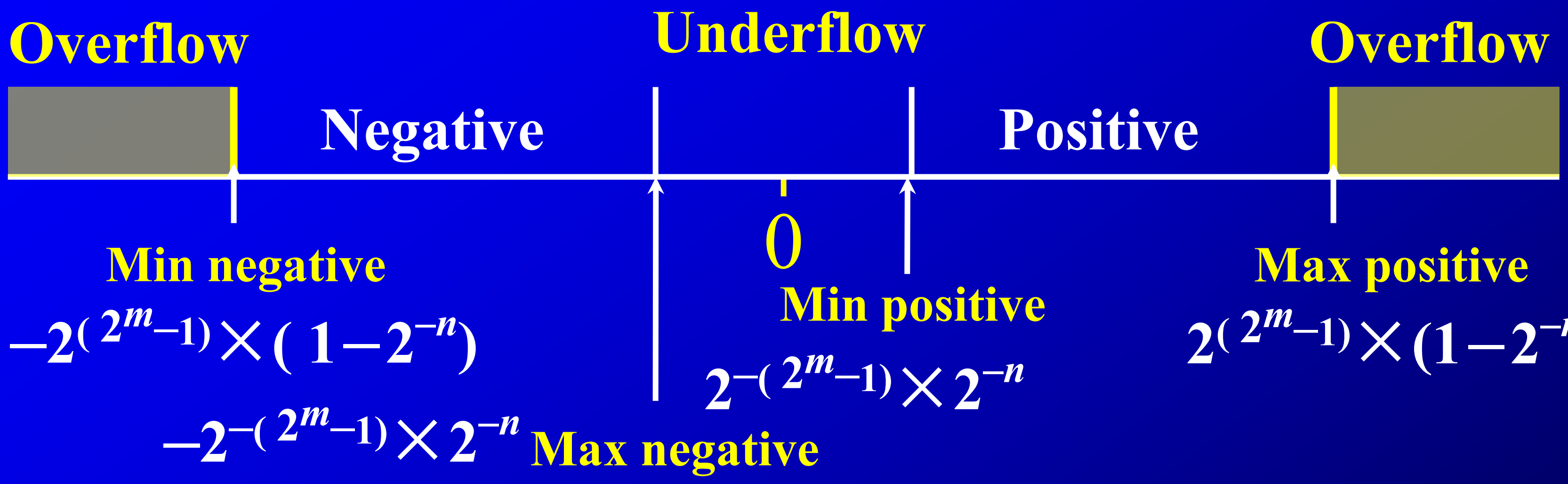
Fixed-point: $x = – 0000\,111010$

Floating norm: $x = – (0.1110100000) \times 2^{0110}$

Fixed point machine	Floating point machine
$[x]_{TF} = 1, 0000111010$	$[x]_{TF} = 0, 0110; 1. 1110100000$
$[x]_{2'} = 1, 1111000110$	$[x]_{2'} = 0, 0110; 1. 0001100000$
$[x]_{1'} = 1, 1111000101$	$[x]_{1'} = 0, 0110; 1. 0001011111$

(c). Exercises

Write the 2's complement form of the floating point number shown in the figure below. Let n = 10, m = 4.



Overflow

Underflow

Overflow

Negative

Positive

0

Min negative

$-2^{(2^m-1)}\times(1-2^{-n})$

Min positive

$2^{-(2^m-1)}\times2^{-n}$

Max positive

$2^{(2^m-1)}\times(1-2^{-n})$

$-2^{-(2^m-1)}\times2^{-n}$ Max negative

Solve:	True value	2's comp
Max positive	$2^{15}\times(1-2^{-10})$	0,1111; 0.1111111111
Min positive	$2^{-15}\times 2^{-10}$	1,0001; 0.0000000001
Max negative	$-2^{-15}\times 2^{-10}$	1,0001; 1.1111111111
Min negative	$-2^{15}\times(1-2^{-10})$	0,1111; 1.0000000001

Machine ZERO

- When the significand of floating-point number is 0, the value is treated as machine ZERO regardless of its exponent

- When the exponent of a floating-point number is equal to or less than the minimum number it represents, it is treated as machine ZERO regardless of the significand value

For example $m = 4$ $n = 10$

When the significand and exponent are expressed in 2's complement, machine ZERO is

×, × × × × ; 0. 0 0 ··· 0

（E=−16） 1, 0 0 0 0 ; ×.×× ··· ×

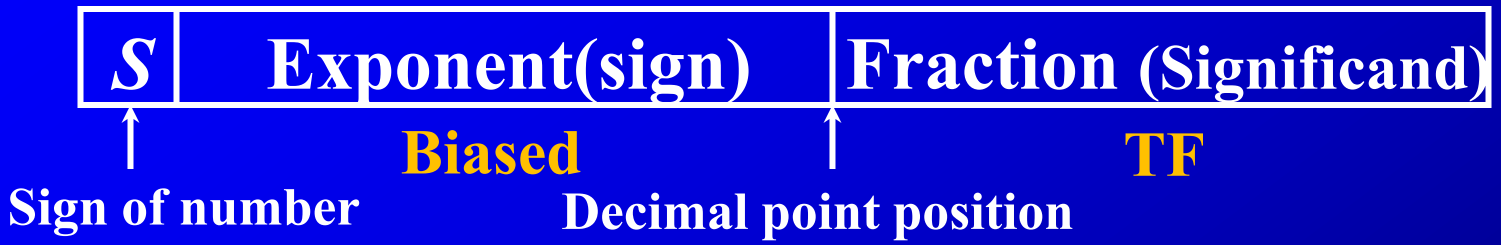
When E is expressed in biased R, S is expressed as 2's comp, machine ZERO is

0, 0 0 0 0; 0. 0 0 ··· 0

Easier in circuit implementation to judge ZERO

(d). IEEE 754 Standard

Institute of Electrical and Electronics Engineers(电气与电子工程师协会)



S | Exponent(sign) | Fraction (Significand)

Biased

TF

Sign of number

Decimal point position

The highest significant bit of non "0" is "1" (implied)

	S	bias	Exponent	Significand	total
F short	1	7FH	8	23	32
F long	1	3FFH	11	52	64
Temporary	1	3FFFH	15	64	80

https://www.zhihu.com/question/24784136

(d). IEEE 754 Standard

$$Value = (-1)^{S} \times (1.ff \ldots ff) \times 2^{E-127}$$

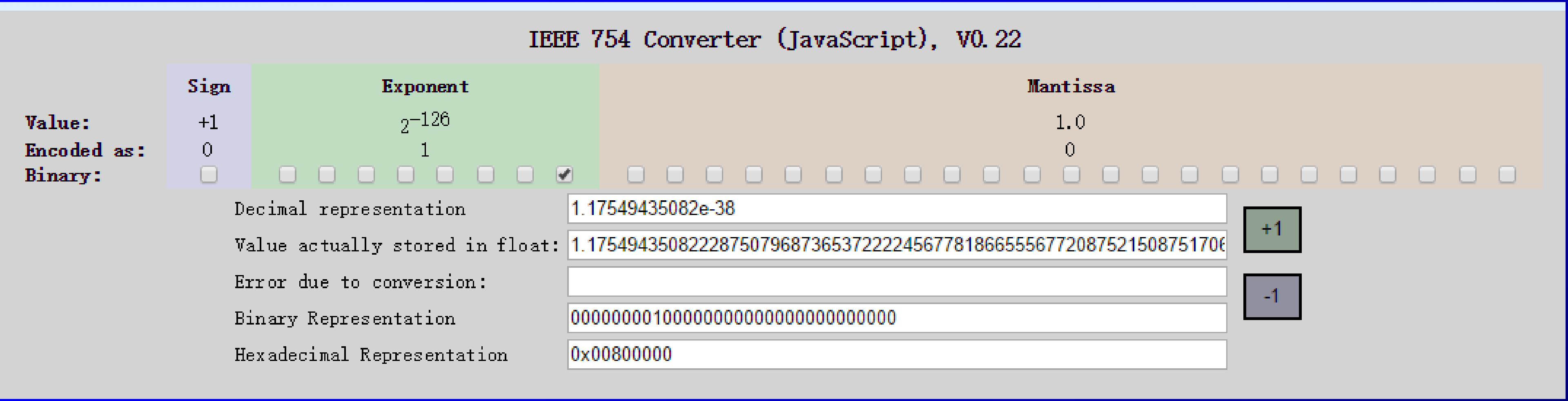
The highest significant bit of non "0" is "1" (implied)

	S	Exponent	Significand	total
F short	1	8	23	32

Max +: 3.4028235e+38

Min +: 1.1754944e -38

偏移127，0和255有特殊用处，故有效范围是（1～254）
阶码E范围是1~254　真值是-126~+127
最大正数：(2^(127))*(2-2^-(23))
最小正数: (2^(-126))*(1)



IEEE 754 Converter (JavaScript), V0.22

	Sign	Exponent	Mantissa
Value:	+1	2^{-126}	1.0
Encoded as:	0	1	0
Binary:	□	□ □ □ □ □ □ □ ☑	□ □

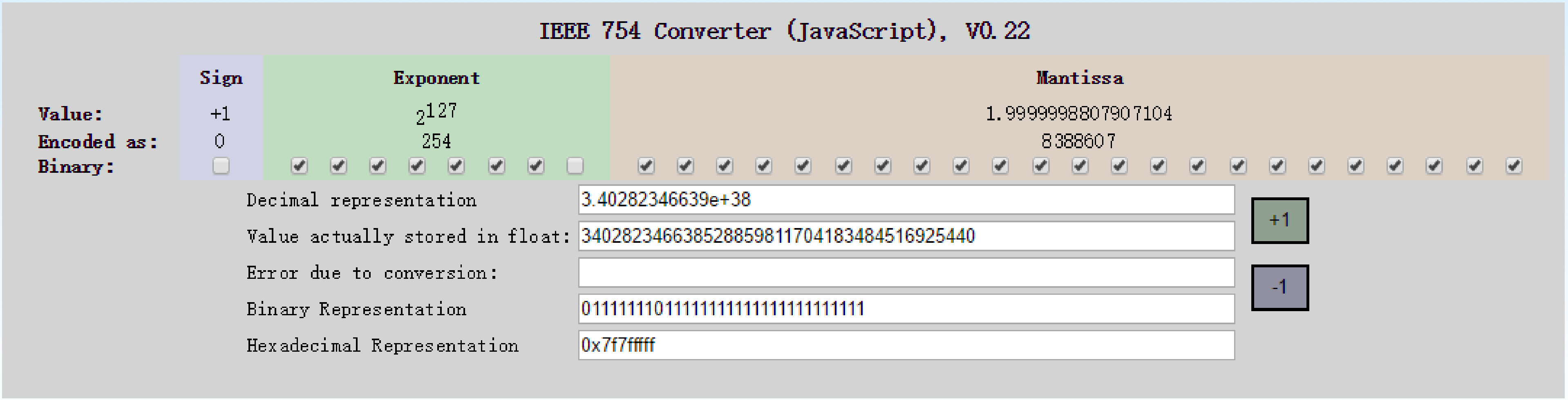
Decimal representation: 1.17549435082e-38

Value actually stored in float: 1.1754943508222875079687365372222456778186655567720875215087517O6

Error due to conversion:

Binary Representation: 00000000100000000000000000000000

Hexadecimal Representation: 0x00800000



IEEE 754 Converter (JavaScript), V0.22

	Sign	Exponent	Mantissa
Value:	+1	2^{127}	1.9999998807907104
Encoded as:	0	254	8388607
Binary:	□	☑ ☑ ☑ ☑ ☑ ☑ ☑ □	☑ ☑

Decimal representation: 3.40282346639e+38

Value actually stored in float: 340282346638528859811704183484516925440

Error due to conversion:

Binary Representation: 01111111011111111111111111111111

Hexadecimal Representation: 0x7f7fffff

https://www.h-schmidt.net/FloatConverter/IEEE754.html

IEEE 754 Standard 178.125

Binary:　10110010.001

Binary floating point representation ：

$1.0110010001 \times 2^{111}$

- **Exponent**

E 8 bits

Bias 7F

- 00000111 + 01111111

$= \underbrace{10000110}_{8}$ Biased'

- **Significand**

$1\ \underbrace{01100100010000000000000}_{23}$

hide

Sign　*E*　*S*

0 10000110 01100100010000000000000

Represents - 27/64 as a 32-bit floating-point normalized number in IEEE754 standard

-27/64 = -11011/1000000 = -0.011011

E（8 Bits）

= -1.1011 $\underline{1\ 0\ 0\ 0\ 0\ 0\ 1\ 0}$(TF-2)

1 1 1 1 1 1 1 0 (2’s)

$\underline{0\ 1\ 1\ 1\ 1\ 1\ 1\ 1}$

Bias 7F

0 1 1 1 1 1 0 1

$1\ 01111101\ \underbrace{10110000000000000000000}_{23}$