CE232 DIGITAL SYSTEM

Topic 6. Signed Number Format

Prepared by Nabila Husna Shabrina

Contact : nabila.husna@umn.ac.id

Subtopic

6.1 Signed Number

6.2 1’s and 2’s Complement

6.3 Fixed Point

6.4 Floating Point IEEE 754

6.1 Signed Number

6.1 Signed Number

Signed number

- Signed number represents negative number
- Computers only know binary, therefore we use signed number to understand how computers represents and computes negative number
- There are 2 method
  - Signed magnitude
  - Complements : one’s complement and two’s complement

6.1 Signed Number

Signed magnitude

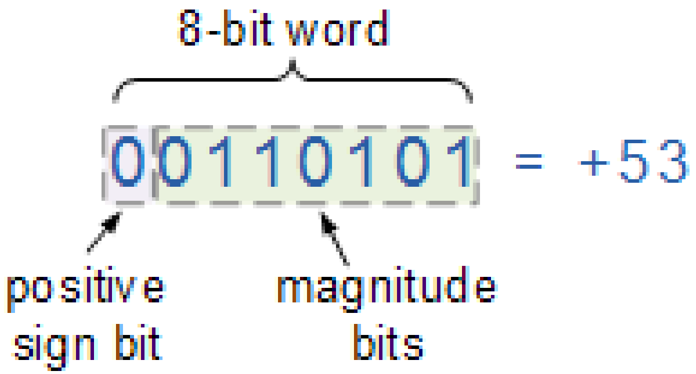
Sign bit	Binary of number

+ represented by zero
- represented by 1

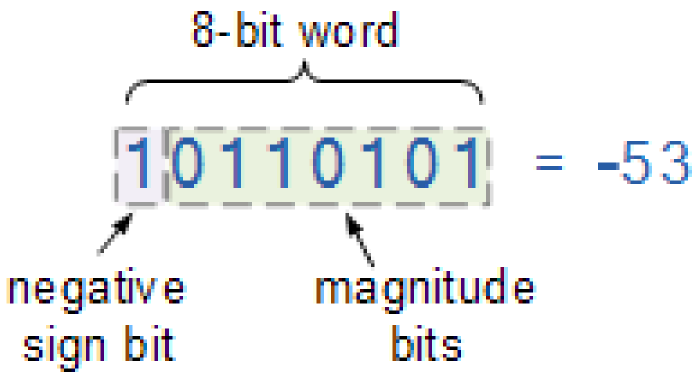
6.1 Signed Number

Example of signed magnitude

Positive number



Negative number



6.1 Signed Number

Disadvantages

- Range only from **$-2^{(n-1)}$ to $+2^{(n-1)}-1$**

Example : the representation for 4 bits signed number (1 for sign bit and 3 bits for magnitude bits) is

$$-2^{(4-1)} - 1 \quad \text{to} \quad +2^{(4-1)} - 1$$

$$-2^{(3)} - 1 \quad \text{to} \quad +2^{(3)} - 1$$

$$-7 \quad \text{to} \quad +7$$

Whereas, in non signed number the range for 4 bit binary is from 0 tp 15

6.1 Signed Number

Example.

-15_{10} as a 6-bit number $\Rightarrow$ 101111_2

$+23_{10}$ as a 6-bit number $\Rightarrow$ 010111_2

-56_{10} as a 8-bit number $\Rightarrow$ 10111000_2

$+85_{10}$ as a 8-bit number $\Rightarrow$ 01010101_2

-127_{10} as a 8-bit number $\Rightarrow$ 11111111_2

6.2 1’s and 2’s Complement

6.2 1's and 2's Complement

- Signed number can also be represented using complements
- There are two complement form
  - 1's complement
  - 2's complement

6.2 1's and 2's Complement

1's Complement

- Complementing each digit in binary number
- Positive number remains the same
- Negative number is represented by complementing every bits
- Range $\mathbf{-2^{(n-1)}}$ **to** $\mathbf{+2^{(n-1)}-1}$

6.2 1's and 2's Complement

Example.

Positive number **$(9)_{10}$ = $(1001)_2$,**

Negative number **$(-9)_{10}$ = $(0110)_2$,**

6.2 1's and 2's Complement

1's complement addition and subtraction

- Subtraction can be implemented in addition form

$$A - B \ = \ A \ + (-B) \ or - (B) + A$$

Steps

- Take 1's complement for subtrahend, add it to the minuend
- If carry is not generated, the result is said to be **negative** & is in 1's complement form, take 1's complement again to get the magnitude of the actual result
- If carry is generated, the result is **positive**, add 1 it to get the actual result

6.2 1’s and 2’s Complement

Example.

Find 3 – 12 using 1’s complement

3 → 0011 ; -12 (using 1’s complement) → 0011

$$\begin{array}{r} 0\,0\,1\,1 \\ \underline{0\,0\,1\,1\,+} \\ 0\,1\,1\,0 \end{array}$$

0 1 1 0 → carry is not generated, take the 1’s complement form to get the magnitude → 1001 so the result is -9

6.2 1’s and 2’s Complement

Example.

Find $54 - 72$ using 1’s complement

54 → 0110110 ; -72 (using 1’s complement) → 0110111

$$\begin{array}{r} 0\,1\,1\,0\,1\,1\,0 \\ \underline{0\,1\,1\,0\,1\,1\,1\,+} \\ 1\,1\,0\,1\,1\,0\,1 \end{array}$$

$1\,1\,0\,1\,1\,0\,1$ → carry is not generated, take the 1’s complement form to get the magnitude → 0010010, so the result is -18

6.2 1's and 2's Complement

Example.

Find 12 – 3 using 1's complement

-3 (using 1's complement) → 1100 ; 12 → 1100

$$\begin{array}{r} 1\,1\,0\,0 \\ \underline{1\,1\,0\,0\,+} \\ 1\,1\,0\,0\,0 \end{array}$$ → carry is generated

$$\begin{array}{r} \underline{\qquad\quad 1\,+} \\ 1\,0\,0\,1 \end{array}$$ → the result is 9

6.2 1's and 2's Complement

Example.

Find $48 - 28$ using 1's complement

48 → 110000 ; -28 (using 1's complement) → 100011

$$\begin{array}{r} 1\,1\,0\,0\,0\,0 \\ \underline{1\,0\,0\,0\,1\,1\,+} \\ 1\,0\,1\,0\,0\,1\,1 \end{array}$$

→ carry is generated, add it to the result

$$\begin{array}{r} \underline{1\,+} \\ 0\,1\,0\,1\,0\,0 \end{array}$$

→ the answer is 20

6.2 1’s and 2’s Complement

2’s Complement

- Complementing each digit in binary number and add by 1
- Positive number remains the same
- Negative number is represented by complementing every bits and adding 1 to the complemented bit
- Range **$-2^{(n-1)}$ to $+2^{(n-1)}-1$**
- The main difference between 1′ s complement and 2′ s complement is that 1′ s complement has two representations of 0, while 2’s complement only have 1 representation for 0

6.2 1's and 2's Complement

Example.

Positive number **$(9)_{10} = (1001)_2$,**

Negative number **$(-9)_{10} = (0111)_2$,**

6.2 1's and 2's Complement

2's complement addition and subtraction

- Subtraction can be implemented in addition form

$$A - B \ = \ A \ + (-B) \ or - (B) + A$$

Steps

- Take 2's complement for subtrahend, add it to the minuend
- If carry is not generated, the result is said to be **negative** & is in 2's complement form. Take 2's complement again to get the magnitude of the actual result
- If carry is generated, discard the carry

6.2 1's and 2's Complement

Example.

Find 48 – 28 using 2's complement

48 → 110000 ; -28 (using 2's complement) → 100100

$$\begin{array}{r} 1\,1\,0\,0\,0\,0 \\ \underline{1\,0\,0\,1\,0\,0\,+} \\ 1\,0\,1\,0\,1\,0\,0 \end{array}$$

→ carry is generated, discard the carry, the answer is 010100 (20)

6.2 1's and 2's Complement

Example.

Find $(111000)_2 - (101001)_2$

$(111000)_2 - (101001)_2 = 56 - 41$

2's complement Of 41 → 010111

$$
\begin{array}{r}
1\ 1\ 1\ 0\ 0\ 0 \\
\underline{0\ 1\ 0\ 1\ 1\ 1\ +} \\
1\ 0\ 0\ 1\ 1\ 1\ 1
\end{array}
$$

→ carry is generated, discard the carry, the answer is 001111 (15)

6.2 1's and 2's Complement

Example.

Find $54 - 72$ using 2's complement

54 → 0110110 ; -72 (using 2's complement) → 0111000

$$
\begin{array}{l}
0\,1\,1\,0\,1\,1\,0 \\
\underline{0\,1\,1\,1\,0\,0\,0\,+} \\
1\,1\,0\,1\,1\,1\,0
\end{array}
$$

→ carry is not generated, take the 2's complement form to get the magnitude → 0010010, so the result is -18

6.3 Fixed Point

6.3 Fixed Point

- Digital system only have 0 and 1
- However, in real world we use fractional number such as 3.12, 3,5, 11,8 etc.
- There are two representation :
  - Fixed point
  - Floating point
- **Fixed point** representation has fixed number of bits for integer part and for fractional part

6.3 Fixed Point

- There are three parts of a fixed-point number representation: the sign field, integer field, and fractional field

6.3 Fixed Point

Example.

Represent fixed point of unsigned binary number 0110110 using 4 integer bits and 3 fractional bits

$$\underbrace{0\,1\,1\,0}_{6}\,.\,\underbrace{1\,1\,0}_{0.75}$$

6.3 Fixed Point

Example.

Represent (-7.5) using 8 bit binary representation with 4 digits integer and 4 fraction bit (using 2's complement)

$$(7{,}5)_{10} \rightarrow (111.1)_2 \rightarrow (0111.1000)$$

$$\Big\downarrow \; 2's\ complement$$

$$1000.1000$$

6.3 Fixed Point

Example.

Compute 0.75 +(-0.625) using 8 bits fixed point number

$(0.75)_{10} \rightarrow (0000.1100)_2$

$(0.625)_{10} \rightarrow (0000.1010)_2$

$\downarrow 2's\ complement$

$(1111.0110)_2$

$$
\begin{array}{r}
0000.1100 \\
\underline{\quad 1111.0110 +} \\
0000.0010
\end{array}
$$

6.4 Floating Point IEEE 754

6.4 Floating Point IEEE 754

- IEEE Standard 754 floating point is the most common representation today for real numbers on computers
- Makes particularly efficient use of the computer to represent extremely large or small value
- Recall scientific notation format
  - a value whose magnitude is in the range of $1 \le n < 10$
  - a power of 10
  - 3498523 is written as 3.498523×10^{6}
  - -0.0432 is written as -4.32×10^{-2}

6.4 Floating Point IEEE 754

- In Binary floating point the format will be written in
  - a value whose magnitude is in the range of $1 \leq n < 2$
  - a power of 2
  - -6.84 is written as -1.71×2^{2}
  - 0.05 is written as 1.6×2^{-5}

6.4 Floating Point IEEE 754

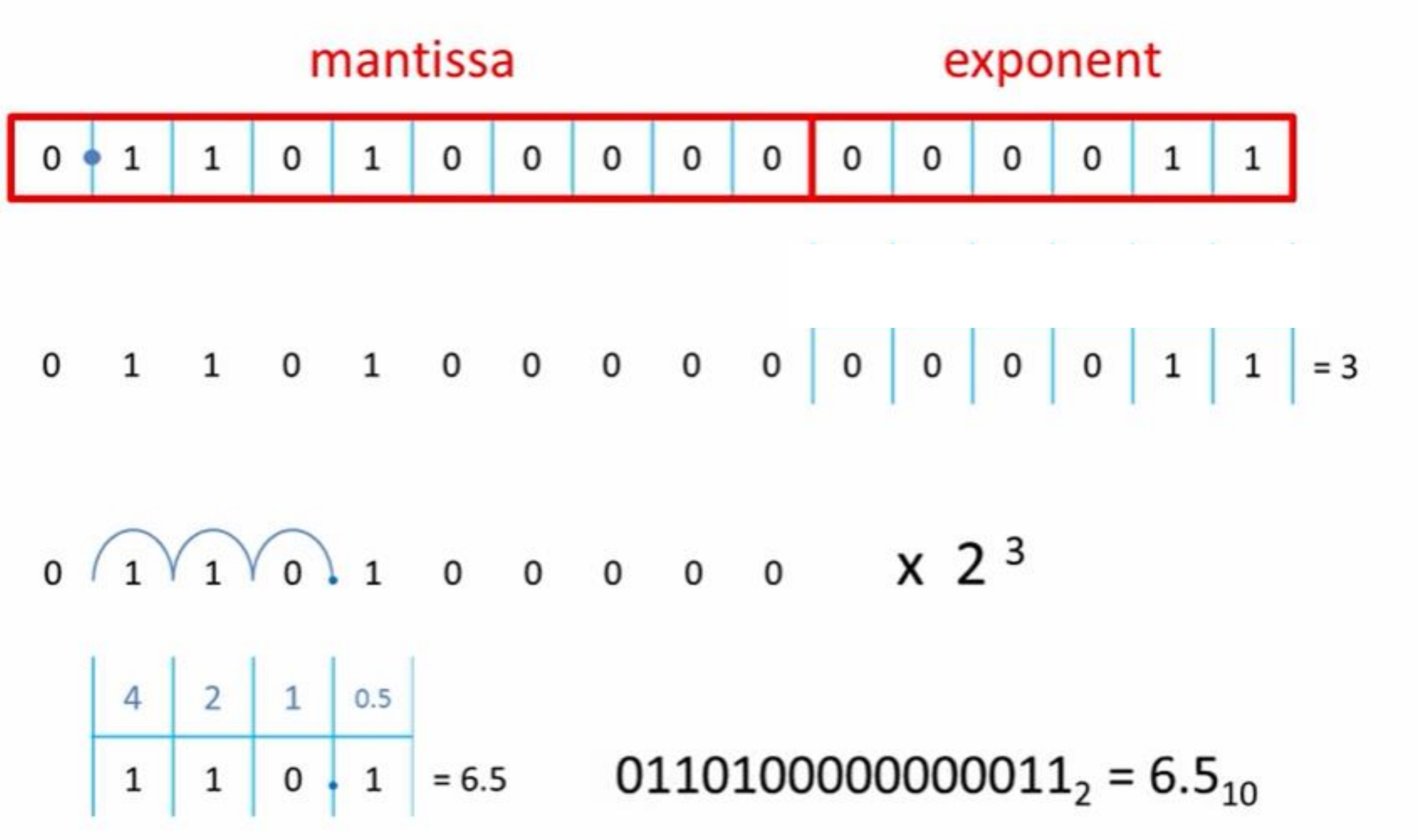
- It has 3 components
  - The sign : 0 represents a positive number while 1 represents a negative number
  - The exponent : to represent both positive and negative exponents.
  - The mantissa : The mantissa is part of a number in scientific notation or a floating-point number

mantissa exponent

$$6.02 \times 10^{23}$$

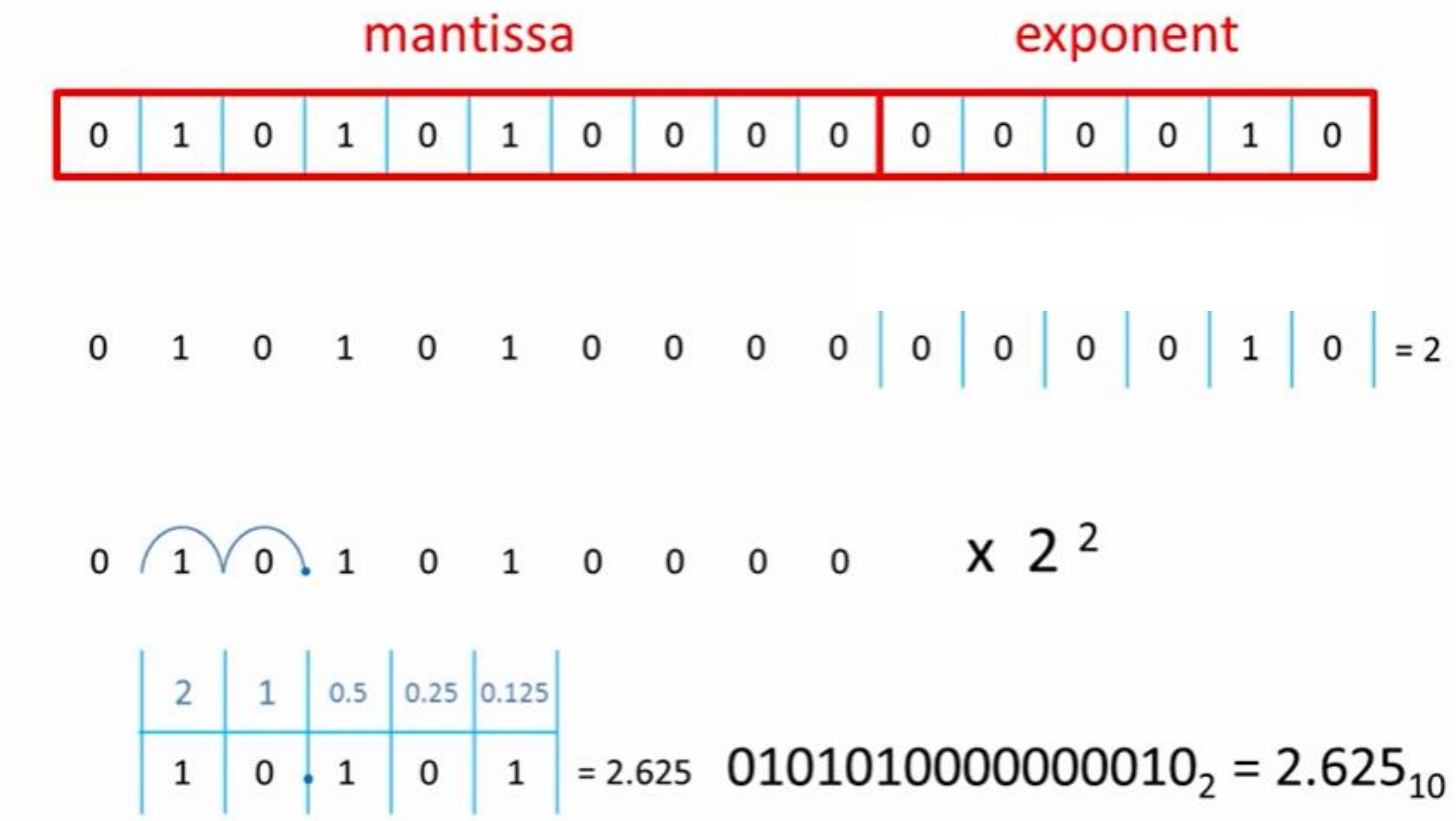
6.4 Floating Point IEEE 754

Example.



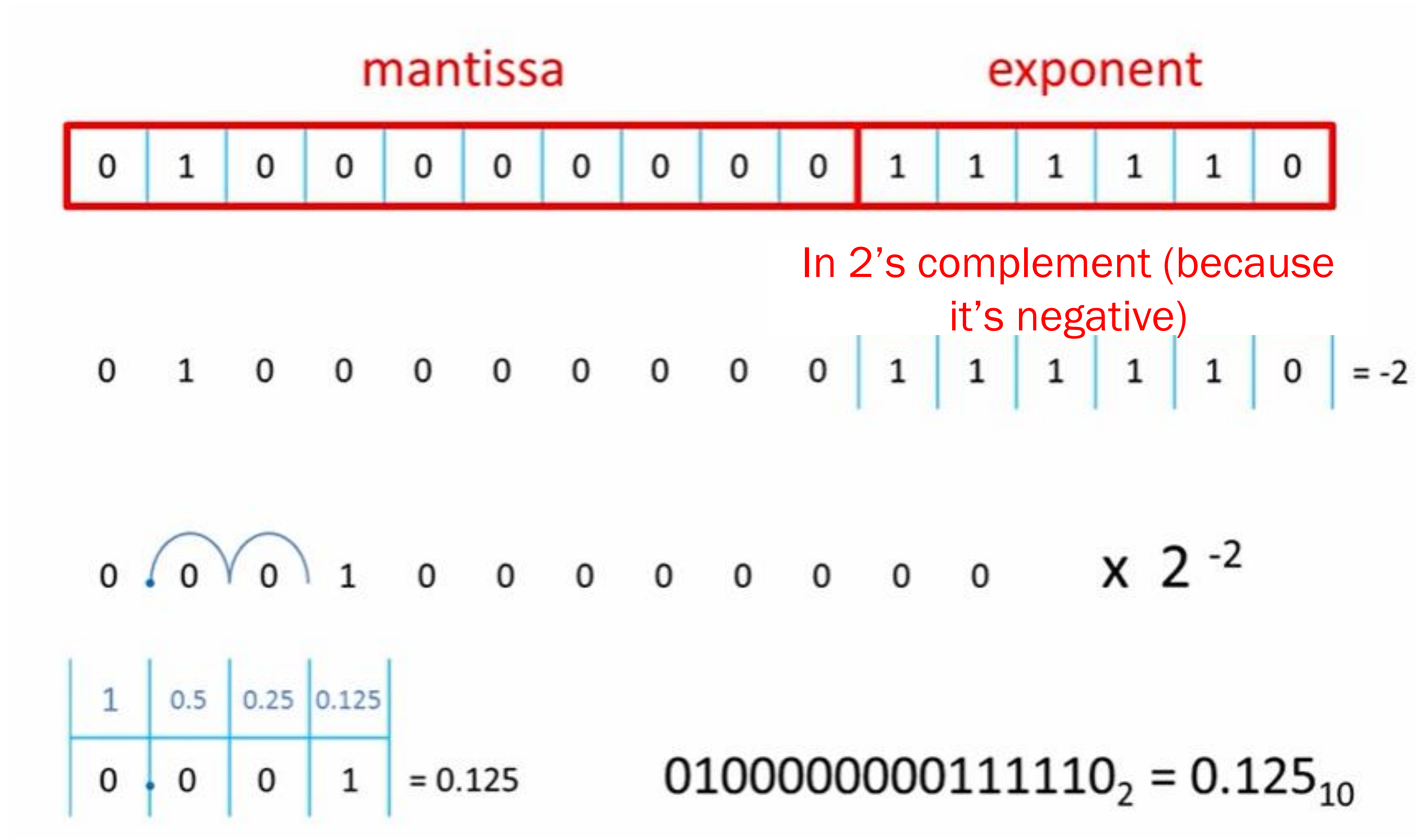
6.4 Floating Point IEEE 754

Example.

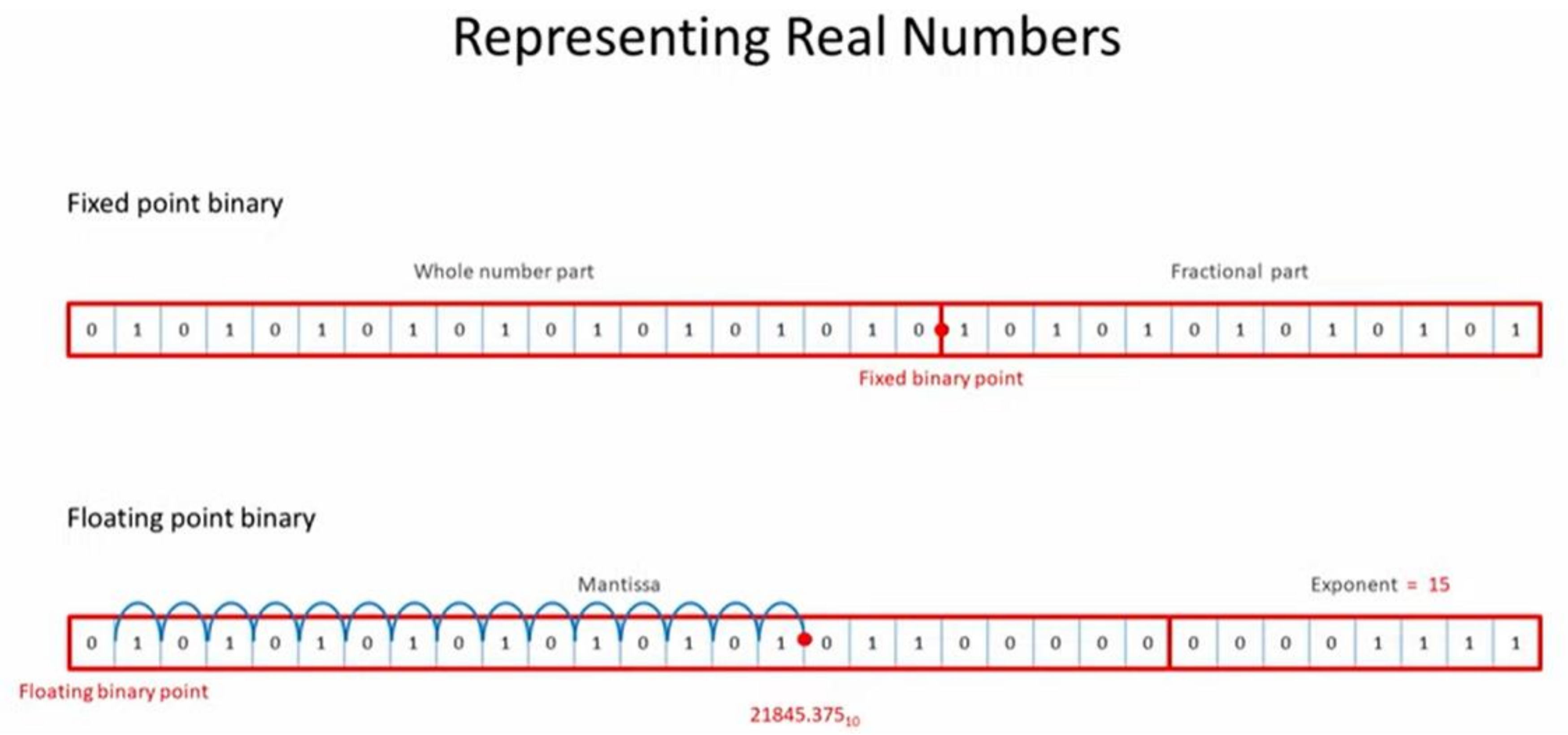


6.4 Floating Point IEEE 754

Example.



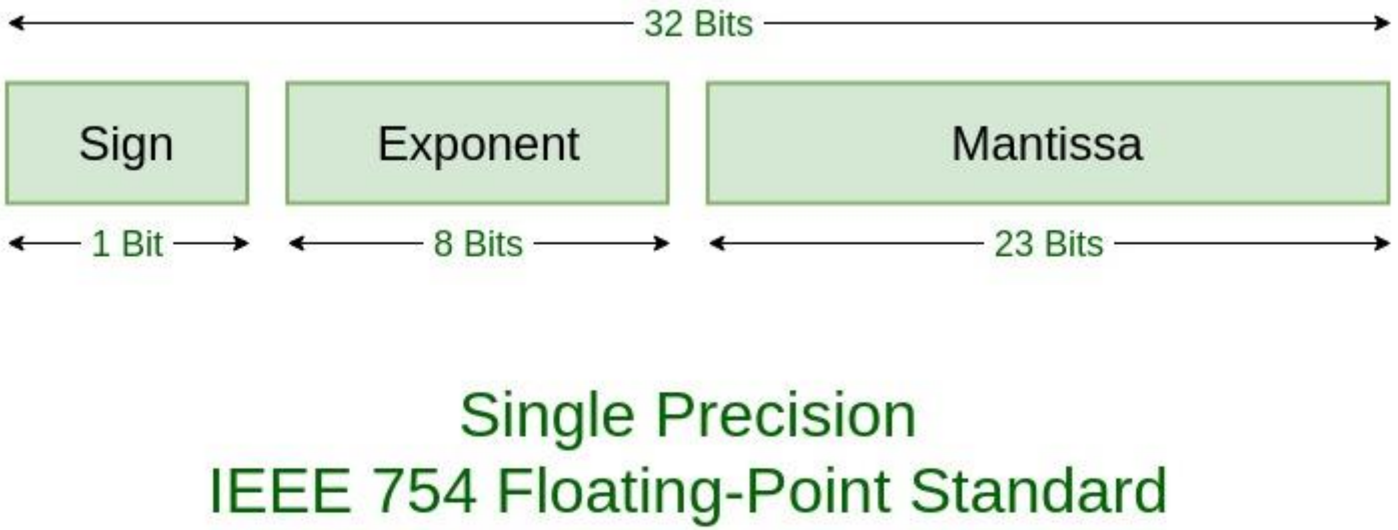
6.4 Floating Point IEEE 754



6.4 Floating Point IEEE 754

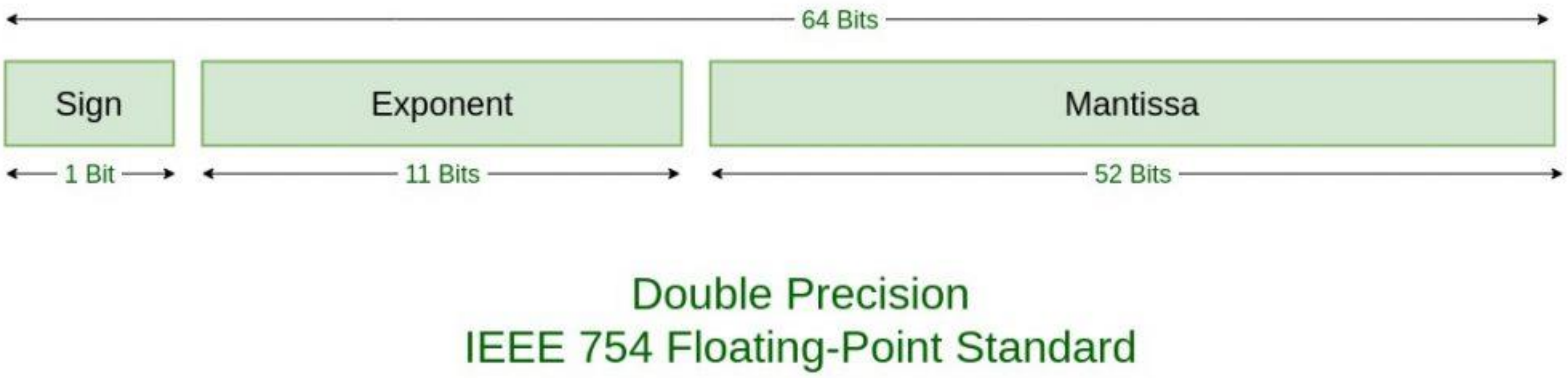
IEEE 754 Standard → **Standard format for floating point**

Single precision



6.4 Floating Point IEEE 754

Double precision



6.4 Floating Point IEEE 754

The difference

IEEE 754 Format	Sign	Exponent	Mantissa	Exponent Bias
32 bit single precision	1 bit	8 bits	23 bits (+ 1 not stored)	$2^{(8-1)} - 1 = 127$
64 bit double precision	1 bit	11 bits	52 bits (+ 1 not stored)	$2^{(11-1)} - 1 = 1023$

6.4 Floating Point IEEE 754

Decimal to IEEE 754 Conversion

- Determine the sign bit
- Convert to pure binary
- Normalize to determine the mantissa and the unbiased exponent by placing the binary point after leftmost 1
- Determine the biased exponent by adding 127 then converting to an unsigned binary integer
- Remove the leading 1 from the mantissa by removing the leftmost 1
- Write the actual result in 32- or 64-bit format

6.4 Floating Point IEEE 754

Example.

Convert 19.25 into IEEE 754 standard 32-bit floating-point binary

Step 1. Determine the sign bit

Positive number → sign bit = 0

Step 2. Convert to pure binary

		16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125
19.59375_{10}	=	1	0	0	1	1 •	1	0	0	1	1

6.4 Floating Point IEEE 754

Step 3. Normalize to determine the mantissa and the unbiased exponent

$$1\ 0\ 0\ 1\ 1\ .\ 1\ 0\ 0\ 1\ 1 = 1.001110011 \times 2^4$$

Step 4. Determined the biased component

$$4 + 127 = 131_{10} = 10000011_2$$

Step 5. Remove the leading 1 from mantissa

$$1.001110011 = 001110011$$

6.4 Floating Point IEEE 754

Step 6. Write the result in 32 format

Sign bit	Exponent								Mantissa																						
0	1	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4 Floating Point IEEE 754

Example.

Convert -123.3 into IEEE 754 standard 32-bit floating-point binary

Step 1. Determine the sign bit

Sign bit = 1

Step 2. Convert to pure binary

$123.3_{10} = 1111011.010011001100110\overline{1001}\ldots_2$

123	÷	2	=	61	remainder	1
61	÷	2	=	30	remainder	1
30	÷	2	=	15	remainder	0
15	÷	2	=	7	remainder	1
7	÷	2	=	3	remainder	1
3	÷	2	=	1	remainder	1
1	÷	2	=	0	remainder	1

0.3	x	2	=	0.6	0
0.6	x	2	=	1.2	1
0.2	x	2	=	0.4	0
0.4	x	2	=	0.8	0
0.8	x	2	=	1.6	1
0.6	x	2	=	1.2	1
0.2	x	2	=	0.4	0

6.4 Floating Point IEEE 754

Step 3. Normalize to determine the mantissa and the unbiased exponent

$$1111011.010011001\overline{1001}\ldots = 1.111011010011001\overline{1001}\ldots \times 2^6$$

Step 4. Determined the biased component

$$6 + 127 = 133_{10} = 10000101_2$$

Step 5. Remove the leading 1 from mantissa

$$1.111011010011001\overline{1001}\ldots = 11101101001100110011001$$

6.4 Floating Point IEEE 754

Round mantissa up if necessary

```
11101101001100110011001
                    + 1
-----------------------
11101101001100110011010
```

Step 6. Write the result in 32 format

1 10000101 11101101001100110011010

6.4 Floating Point IEEE 754

IEEE 754 to Decimal Conversion

- Determine the sign bit
- Determine the exponent in decimal
- Remove the exponent bias by subtracting 127
- Convert the mantissa to decimal
- Add 1 to the mantissa and include the sign
- Calculate the final result

6.4 Floating Point IEEE 754

Example.

Convert 0 10000100 11010100000000000000000

Step 1. Determine the sign bit

0 → positive

Step 2. Determine the exponent in decimal

$(100000100)_2 = (132)_{10}$

Step 3. Remove the exponent bias

$$132 - 127 = 5$$

6.4 Floating Point IEEE 754

Step 4. Convert the mantissa to decimal

	0.5	0.25	0.125	0.0625	0.03125	0.015625
•	1	1	0	1	0	1

0.5 + 0.25 + 0.0625 + 0.015625 = 0.828125

Step 5. Add 1 to the mantissa and include in the sign

0.828125 + 1 = 1.828125

Step 6. Calculate the final result

$1.828125 \times 2^5 = 58.5$

6.4 Floating Point IEEE 754

Example.

Convert 1 00001001 00011001000000000000000

Step 1. Determine the sign bit

1 → negative

Step 2. Determine the exponent in decimal

$(00001001)_2 = (9)_{10}$

Step 3. Remove the exponent bias

$$9 - 127 = -118$$

6.4 Floating Point IEEE 754

Step 4. Convert the mantissa to decimal

0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	0.00390625
• 0	0	0	1	1	0	0	1

0.0625 + 0.03125 + 0.00390625 = 0.09765625

Step 5. Add 1 to the mantissa and include in the sign

0.09765625 + 1 = 1.09765625 → -1.09765625 (because the sign is –)

Step 6. Calculate the final result

$$-1.09765625\text{x } 2^{-118} = -3.30313912581062 \ldots x2^{-36}$$

6.4 Floating Point IEEE 754

Reserved value

Exponent Value	Mantissa	Represents
11111111	All zeros	Infinity (∞)
11111111	Not all zeros	Not a number (NAN)
00000000	All zeros	Zero
00000000	Not all zeros	Subnormal (very small)

References

M. Morris Mano, Digital Design, 5th ed, Prentice Hall, 2012, **Chapter 1**

Next Topic : Combinational Logic Circuit