

---

# Arithmetic Operations

---

SCS 1205

---

Computer Systems

---

# Binary Addition

Addition	Result	Carry
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1

$$\begin{array}{r} \textcolor{red}{1} & \textcolor{red}{1} & \textcolor{red}{1} & \leftarrow \text{Carry} \\ 1 & 1 & 0 & 1 \\ + & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 \end{array}$$

Addition	Result	Carry
$1 + 1 + 1$	1	1

# Binary Subtraction

Subtraction	Result
0 - 0	= 0
0 - 1	= 1 * with borrow
1 - 0	= 1
1 - 1	= 0

$$\begin{array}{r} 101101 \\ - 010111 \\ \hline = 010110 \end{array}$$

\* \* \* (borrow)

# Binary Multiplication

Multiplication	Result
$0 * 0$	0
$0 * 1$	0
$1 * 0$	0
$1 * 1$	1

$$\begin{array}{r} 1 \ 0 \ 1 \ 1 \leftarrow \text{Multiplicand} \\ \times 1 \ 0 \ 1 \ 0 \leftarrow \text{Multiplier} \\ \hline 0 \ 0 \ 0 \ 0 \\ + \ 1 \ 0 \ 1 \ 1 \\ + \ 0 \ 0 \ 0 \ 0 \\ + \ 1 \ 0 \ 1 \ 1 \\ \hline = 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \end{array}$$

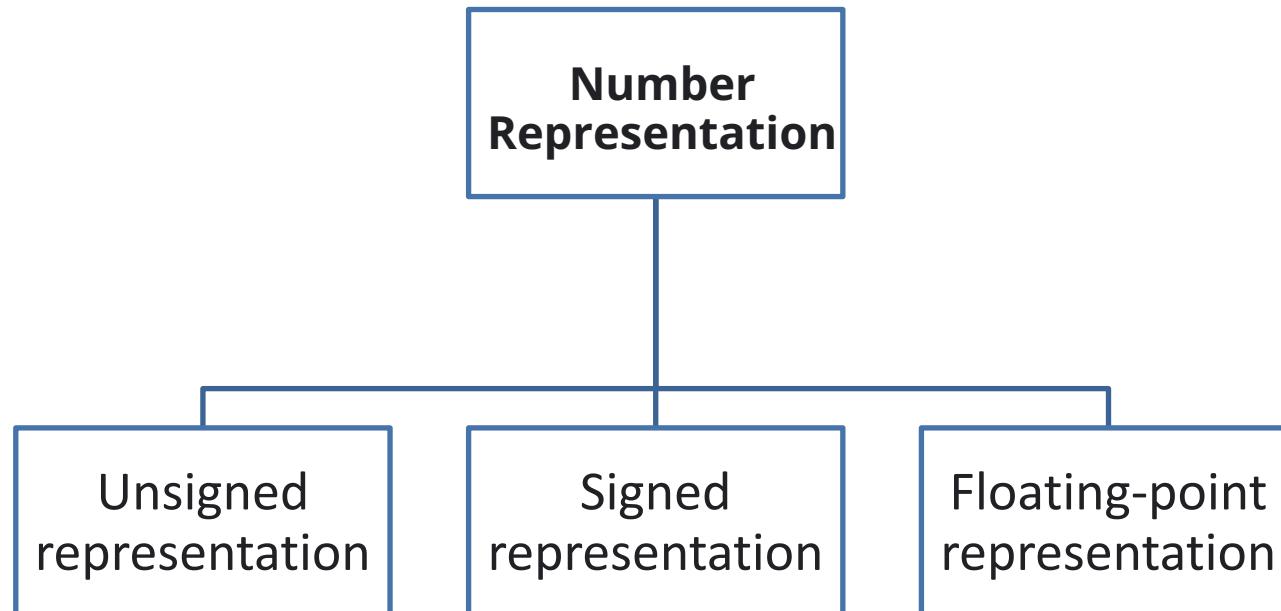
# Binary Division

- Decide whether it is 0 or 1

A binary division diagram. The divisor '1 0 1' is written vertically on the left. The dividend '1 1 0 1 1' is at the top. The quotient '1 0 1' is written above the dividend with an orange arrow pointing to the right labeled 'Quotient'. The first subtraction step shows '1 0 1' subtracted from '1 1 0' resulting in '0 1 1'. The second subtraction step shows '0 0 0' subtracted from '1 1' resulting in '1 1'. The third subtraction step shows '1 0 1' subtracted from '1 1' resulting in '1 0'. An orange arrow points down from the first subtraction result to the second, and another points down from the second to the third, both labeled 'Divisor'. An orange arrow points left from the final result '1 0' labeled 'Remainder'.

$$\begin{array}{r} 1 0 1 \leftarrow \text{Quotient} \\ \begin{array}{r} 1 1 0 1 1 \\ - 1 0 1 \\ \hline 0 1 1 \end{array} \\ \begin{array}{r} 0 1 1 \\ - 0 0 0 \\ \hline 1 1 \end{array} \\ \begin{array}{r} 1 1 \\ - 1 0 1 \\ \hline 1 0 \end{array} \leftarrow \text{Remainder} \end{array}$$

# Number representation

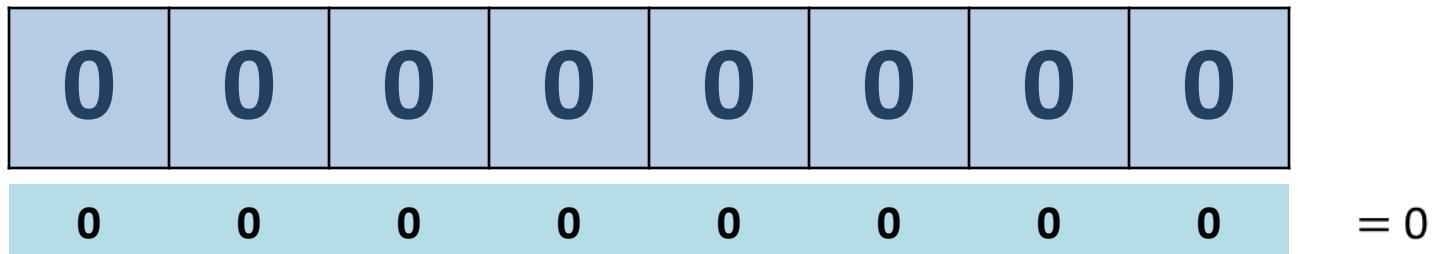


# Unsigned Binary Numbers

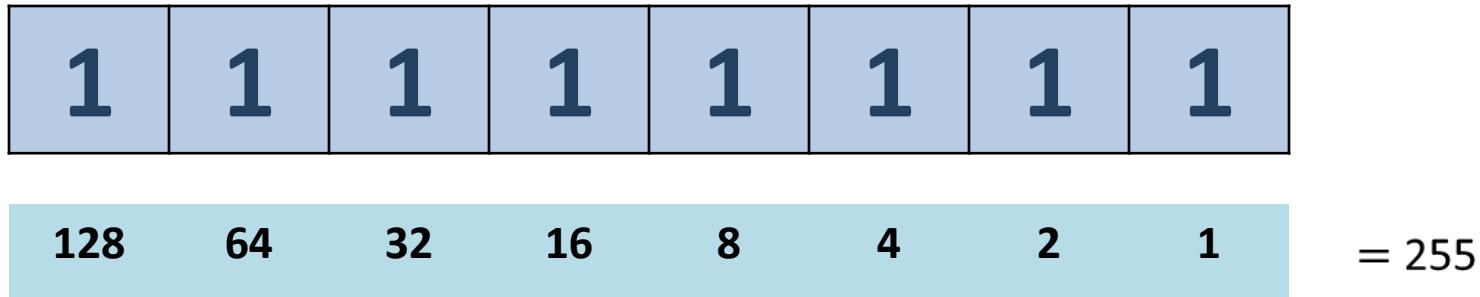
Magnitude is the bits in the pattern which store the size of the number

- All the bits are used for representing magnitude

Smallest



Largest

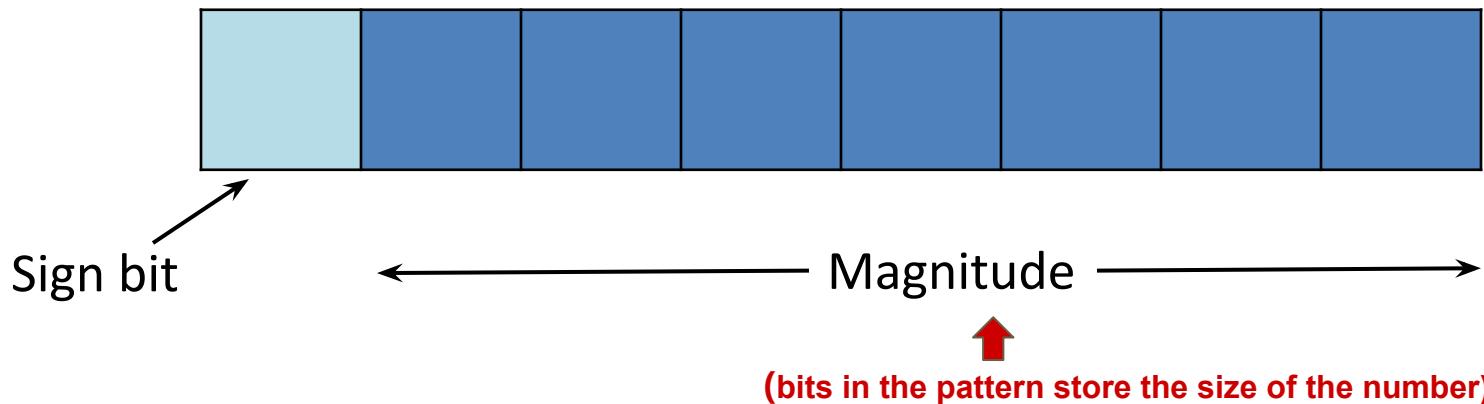


# Signed Binary Numbers

- Left most number is the **sign bit**.

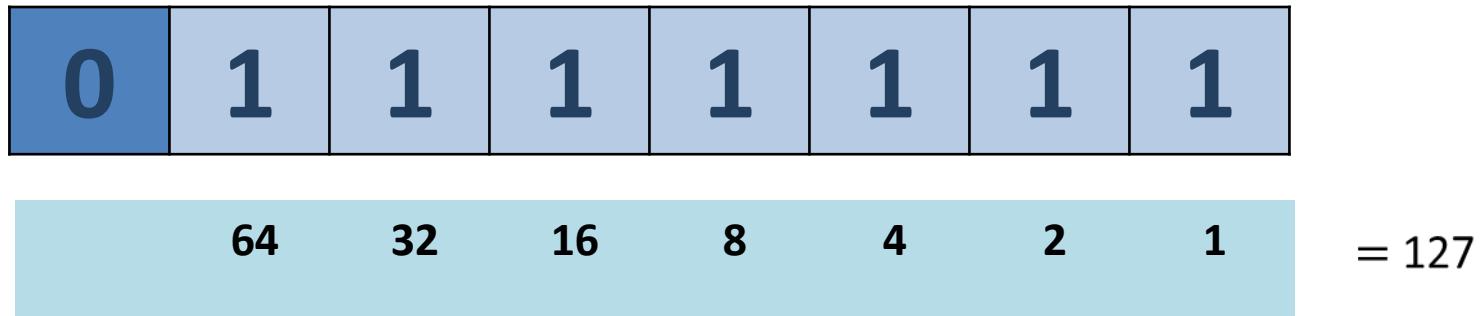
**0** -> positive

**1** -> negative



# Signed Binary Numbers

- The largest magnitude is **127**, which is approximately half of what is for unsigned binary numbers.



# Problems in signed representation

- +0 and -0
- Need to consider both sign and magnitude in arithmetic

$$7 - 4 = ??$$

$$= 7 + (-4)$$

$$= 3$$

$$\begin{array}{r} 00000111 \\ + 10000100 \\ \hline 10001011 \end{array} = -11$$



# One's Complement

A method which we can use to represent negative binary numbers in a signed binary number system.

- One's complement of 1 is 0
- One's complement of 0 is 1

1011010  
01001010 ← One's Complement

# One's Complement

$$7 - 4 = ??$$

$$= 7 + (-4)$$

$$\begin{array}{r} 00000100 \\ \hline 11111011 \end{array} \quad \rightarrow 4$$

one's complement value of 4

$$\begin{array}{r} 1111111 \\ 00000111 \\ + 11111011 \\ \hline 100000010 \\ + 1 \\ \hline 00000011 = 3 \end{array}$$

Carry

7

one's complement of 4

- ❖ In one's complement if there is a carry value generated after the sign bit. it will be added to result.

# One's Complement

$$115 - 27 = ??$$

$$= 115 + (-27)$$

$$00011011 \longrightarrow 27$$

$$11100100$$



one's complement value of 27

$$\begin{array}{r} & 11 \leftarrow \text{Carry} \\ & 01110011 \leftarrow 115 \\ + & 11100100 \leftarrow \text{one's complement of } 27 \\ \hline & 101010111 \\ & + \quad \quad \quad 1 \\ \hline & 01011000 = 88 \\ \hline \end{array}$$

# Two's Complement

- 2's complement of a binary number is 1 added to the 1's complement of the binary number.

00000111 ← Binary Number

11111000 ← One's complement

$$\begin{array}{r} + \quad \quad \quad 1 \\ \hline \end{array}$$

11111001 ← Two's complement

$$\begin{array}{r} \hline \hline \end{array}$$

# Two's Complement

$$7 - 4 = ??$$

$$= 7 + (-4)$$

Binary

$$\rightarrow \underbrace{00000100}_{\text{1's complement}}$$

1's complement  $\rightarrow 11111011$

$$\begin{array}{r} + \\ \hline \end{array} \quad 1$$

2's complement  $\rightarrow 11111100$

11111

Carry

$$00000111$$

$$+ \quad 11111100$$

$$\underline{\underline{100000011}}$$

= 3

# Two's Complement

$$115 - 27 = ??$$

$$= 115 + (-27)$$

00011011

1's complement → 11100100

+1

2's complement → 11100101

11 111      Carry

01110011

+

11100101

---

101011000      = 88

---

---

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1

# Fractions in binary to decimal

<b>1</b>	<b>0</b>	<b>1</b>	.	<b>0</b>	<b>1</b>	<b>1</b>
↓	↓	↓		↓	↓	↓
$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	$2^{-3}$
4	0	1		0	$1/4$	$1/8$

$$4 + 0 + 1 + \frac{1}{4} + \frac{1}{8} = 5 \frac{3}{8}$$

$$101.011_2 = 5.375_{10}$$

# Fractions : binary to decimal

$$\begin{array}{ccccc} 1 & 1 & 1 & . & 0 \end{array} \quad \begin{array}{ccccc} 1 & 1 & 1 & . & 0 \end{array}$$
$$\begin{array}{ccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 4 & 2 & 1 & 0 & \frac{1}{4} \end{array}$$

$$4 + 2 + 1 + \frac{1}{4} = 7\frac{1}{4}$$

$$111.01 = 7\frac{1}{4}$$

# Fractions : decimal to binary

1. Multiply the number by the base (=2)
2. Take the integer on the left (0 or 1) as the coefficient.
3. Take the resultant fraction and repeat the division until resultant is zero

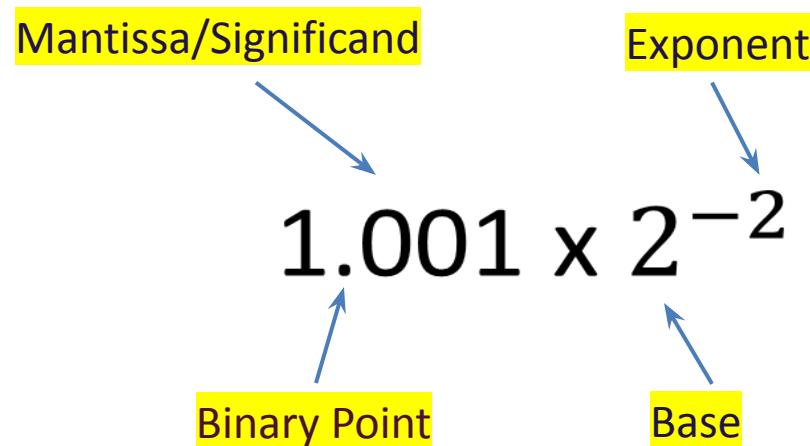
Eg :  $0.750_{10}$

A vertical division diagram for binary conversion. On the left, there is a blue arrow pointing downwards. To its right is a vertical line. To the right of the vertical line is a table with three columns. The first column contains the coefficients (0, 1, 1). The second column contains the multipliers ( $750 * 2$ ,  $500 * 2$ , 000). The third column contains the results (500, 000, 000). A horizontal line separates the first row from the second. The third row has a single value '1' in the first column.

$$\begin{array}{c|cc} & 0 & 750 * 2 \\ \hline & 1 & 500 * 2 \\ & 1 & 000 \end{array}$$

$0.750_{10} = 0.11_2$

# Non-integer Representation (Float point)



# Scientific Notation

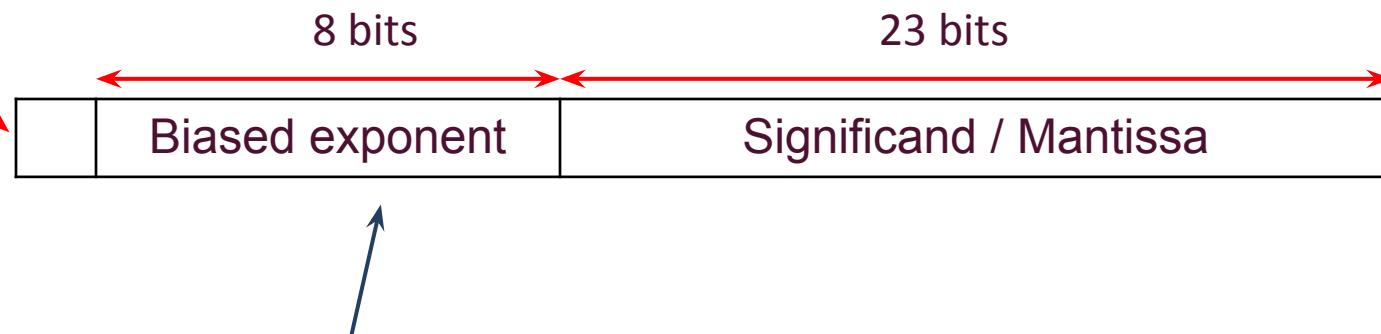
$$144.69 = 1.4469 \times 10^2$$

$$0.01001 = 1.001 \times 2^{-2}$$

# Binary floating point representation

- 32-bit floating point representation
- Also called as single precision.

Sign of  
significand



Binary representation = exponent + bias

For 8 bits -> bias =  $2^7 - 1 = 127$

# Binary floating point representation

Represent 175.625 in 32-bit binary floating point number

Eg : Represent 175.625 in 32-bit binary floating point number

## STEP 01 : Convert the given decimal number to binary

2	175
2	87 - 1
2	43 - 1
2	21 - 1
2	10 - 1
2	5 - 0
2	2 - 1
1	- 0

0	625 * 2
1	250 * 2
0	500 * 2
1	000

$$175.625_{10} = 10101111.101_2$$

Eg : Represent 175.625 in 32-bit binary floating point number

## STEP 02 : Represent binary number in the scientific notation

Binary number → 10101111.101<sub>2</sub>

$$10101111.101 = \underbrace{1.010111101}_{\text{Mantissa}} \times 2^7$$

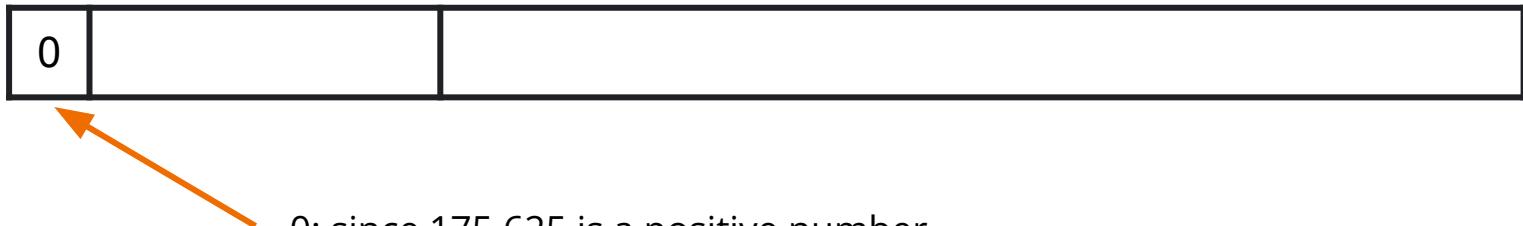
Eg : Represent 175.625 in 32-bit binary floating point number

### STEP 03 : Represent scientific notation in 32-bit binary floating number

- i) Assign the sign bit

$1.0101111101 \times 2^7$

Mantissa



Eg : Represent 175.625 in 32-bit binary floating point number

### **STEP 03 : Represent scientific notation in 32-bit binary floating number**

ii) Find the binary representation of biased exponent

$$1.0101111101 \times 2^7$$

(For a 8 bit exponent → bias = 127)

$$\begin{aligned}\text{Biased exponent} &= \text{bias} + \text{exponent} \\ &= 127 + 7 \\ &= 134\end{aligned}$$

**Binary representation of  $134_{10} = 10000110_2$**

0	10000110	
---	----------	--

Eg : Represent 175.625 in 32-bit binary floating point number

### STEP 03 : Represent scientific notation in 32-bit binary floating number

- iii) Assign the mantissa and fill the rest with zeros

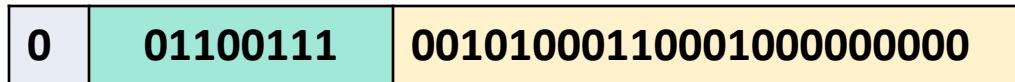
$$1.\underbrace{0101111101}_{\text{Mantissa}} \times 2^7$$

0	1 0 0 0 0 1 1 0	0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
---	-----------------	-------------------------------------------------------------------

# Binary floating point representation

Eg : Represent the following 32-bit binary floating point number in scientific notation.

0	01100111	001010001100010000000000
---	----------	--------------------------



Sign bit = 0;  
**positive**

$$01100111_2 = 103_{10}$$

**Bias adjustment** =  $103 - 127$   
= -24

**Significand :**

$$\begin{aligned}
 &= 1 + (0*2^{-1}) + (0*2^{-2}) + (1*2^{-3}) + (0*2^{-4}) + (1*2^{-5}) + \dots \\
 &= 1 + 0 + 0 + \frac{1}{8} + 0 + \frac{1}{32} \dots \\
 &= 1 + 0.1592407 \\
 &= \mathbf{1.1592407}
 \end{aligned}$$

$$1.1592407 \times 2^{-24} \xleftarrow{\text{Simplify this to get the decimal scientific notation}}$$

**Scientific Notation :  $6.909613013 \times 10^{-8}$**

# Thank You!!