

# Number System (1A)

---

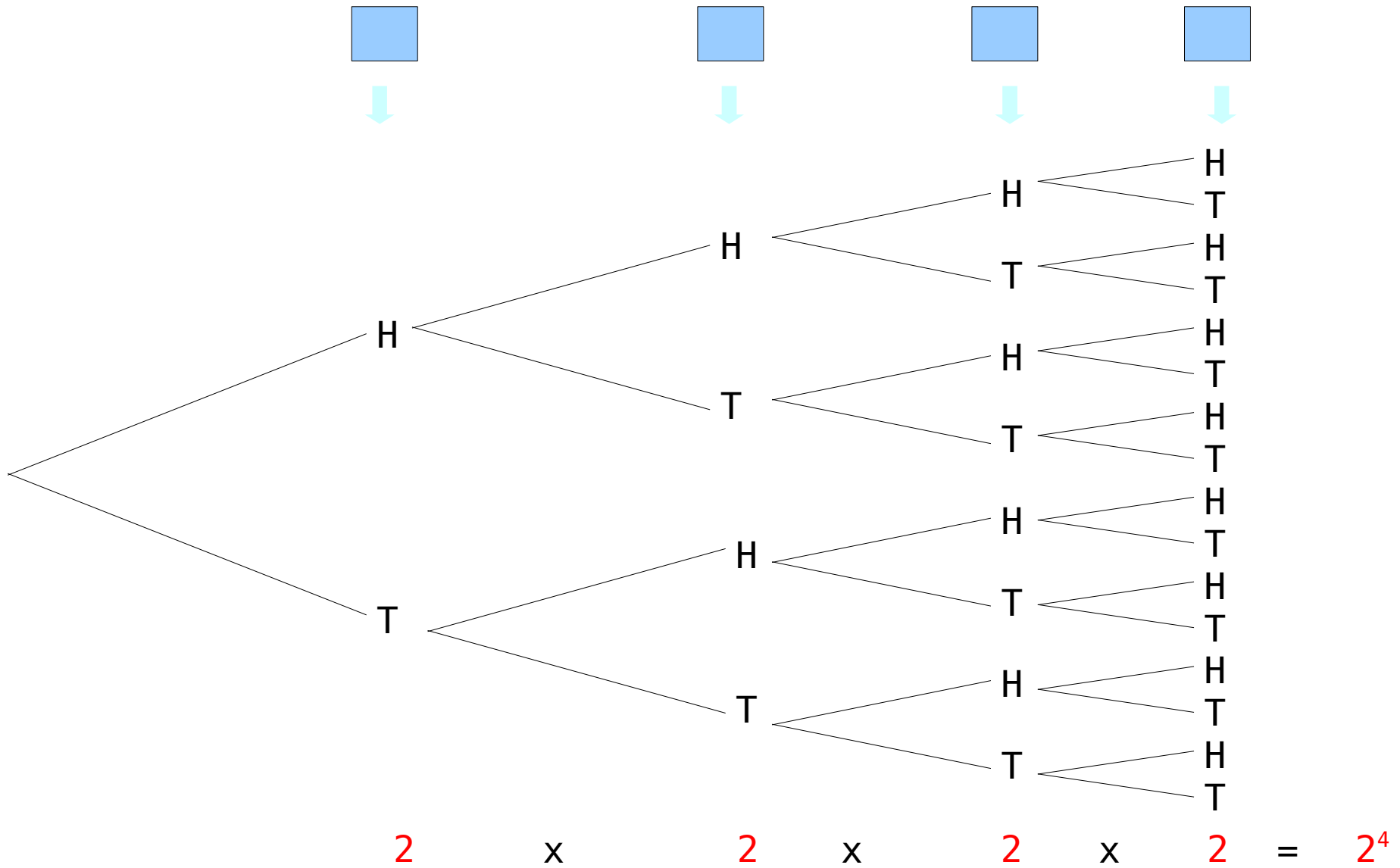
Copyright (c) 2009 - 2016 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

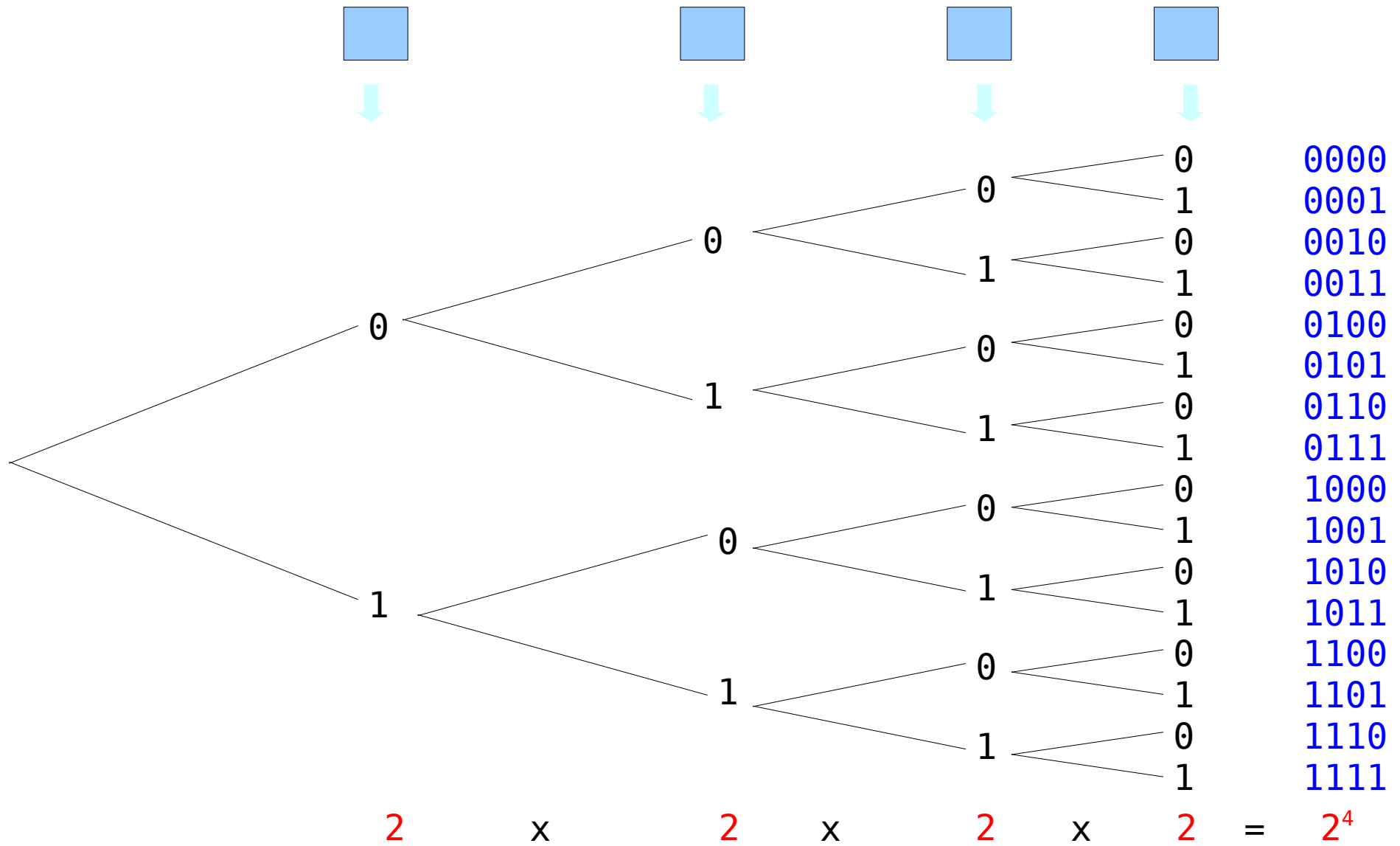
Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice/LibreOffice.

# Coin Tossing

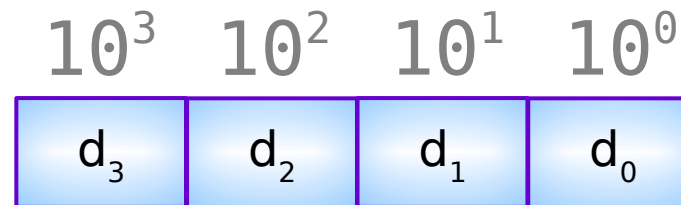


# Coin Tossing and Binary Number

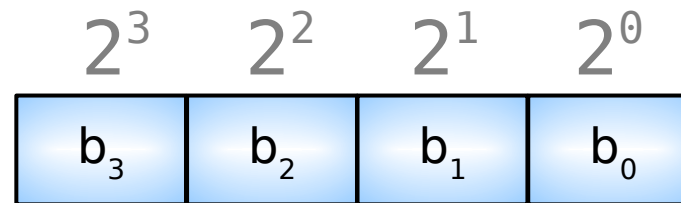


# Radix Number Systems

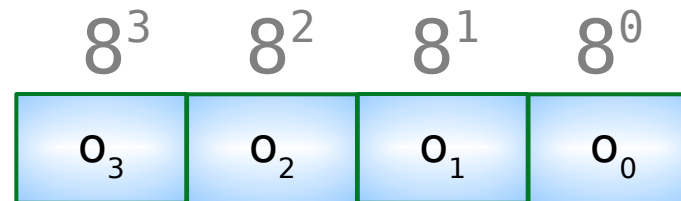
Decimal



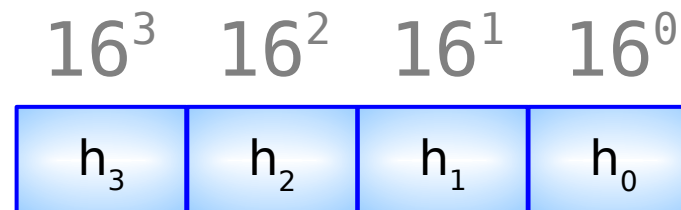
Binary



Octal



Hexadecimal



# Conversion to decimal numbers

Decimal

$$d_3 \cdot 10^3 + d_2 \cdot 10^2 + d_1 \cdot 10^1 + d_0 \cdot 10^0$$

Binary

$$b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

Octal

$$o_3 \cdot 8^3 + o_2 \cdot 8^2 + o_1 \cdot 8^1 + o_0 \cdot 8^0$$

Hexadecimal

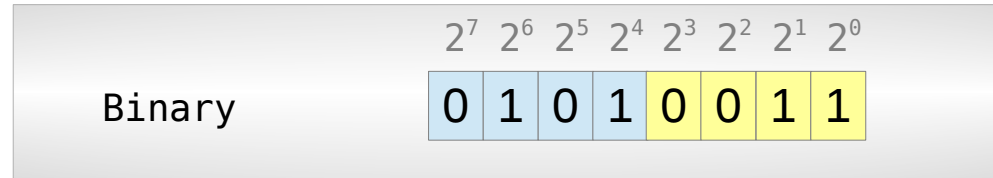
$$h_3 \cdot 16^3 + h_2 \cdot 16^2 + h_1 \cdot 16^1 + h_0 \cdot 16^0$$

# Number Systems [0..15]

$10^1 10^0$		$2^3 2^2 2^1 2^0$				$16^0$	$8^1 8^0$	
radix=10	0	radix=2				radix=16	radix=8	
	1							
	2							
	3							
	4							
	5							
	6							
	7							
	8							
	9							
	1 0							
	1 1							
	1 2							
	1 3							
	1 4							
	1 5							
Decimal		Binary				Hexadecimal	Octal	

# Binary and Hexadecimal Numbers

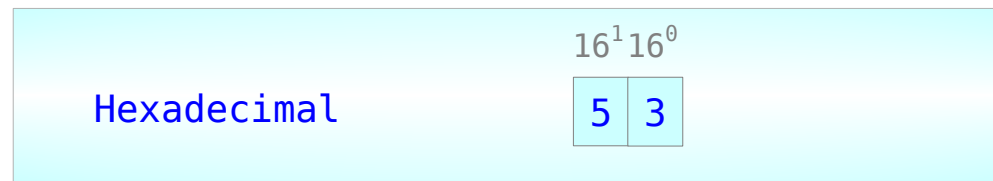
	$2^3$	$2^2$	$2^1$	$2^0$		$16^0$
radix=2	0	0	0	0	radix=16	0
	0	0	0	1		1
	0	0	1	0		2
	0	0	1	1		3
	0	1	0	0		4
	0	1	0	1		5
	0	1	1	0		6
	0	1	1	1		7
	1	0	0	0		8
	1	0	0	1		9
	1	0	1	0		A
	1	0	1	1		B
	1	1	0	0		C
	1	1	0	1		D
	1	1	1	0		E
	1	1	1	1		F
	Binary					Hexadecimal



$$\underbrace{(2^3 \ 2^2 \ 2^1 \ 2^0)}_{16^0} \times 2^4 + \underbrace{2^3 \ 2^2 \ 2^1 \ 2^0}_{16^0}$$

$$\underbrace{0 \ 1 \ 0 \ 1}_{16^0} \times 2^4 + \underbrace{0 \ 0 \ 1 \ 1}_{16^0}$$

$$\underbrace{5}_{16^0} \times 16^1 + \underbrace{3}_{16^0}$$



Decimal

$$5 \times 16^1 + 3 = 83$$



# Bit Patterns

	$2^3$	$2^2$	$2^1$	$2^0$		$16^0$		$8^1$	$8^0$
radix=2	0	0	0	0	radix=16	0	radix=8	0	0
	0	0	0	1		1		0	1
	0	0	1	0		2		0	2
	0	0	1	1		3		0	3
	0	1	0	0		4		0	4
	0	1	0	1		5		0	5
	0	1	1	0		6		0	6
	0	1	1	1		7		0	7
	1	0	0	0		8		1	0
	1	0	0	1		9		1	1
	1	0	1	0		A		1	2
	1	0	1	1		B		1	3
	1	1	0	0		C		1	4
	1	1	0	1		D		1	5
	1	1	1	0		E		1	6
	1	1	1	1		F		1	7
Binary					HEX	OCT			

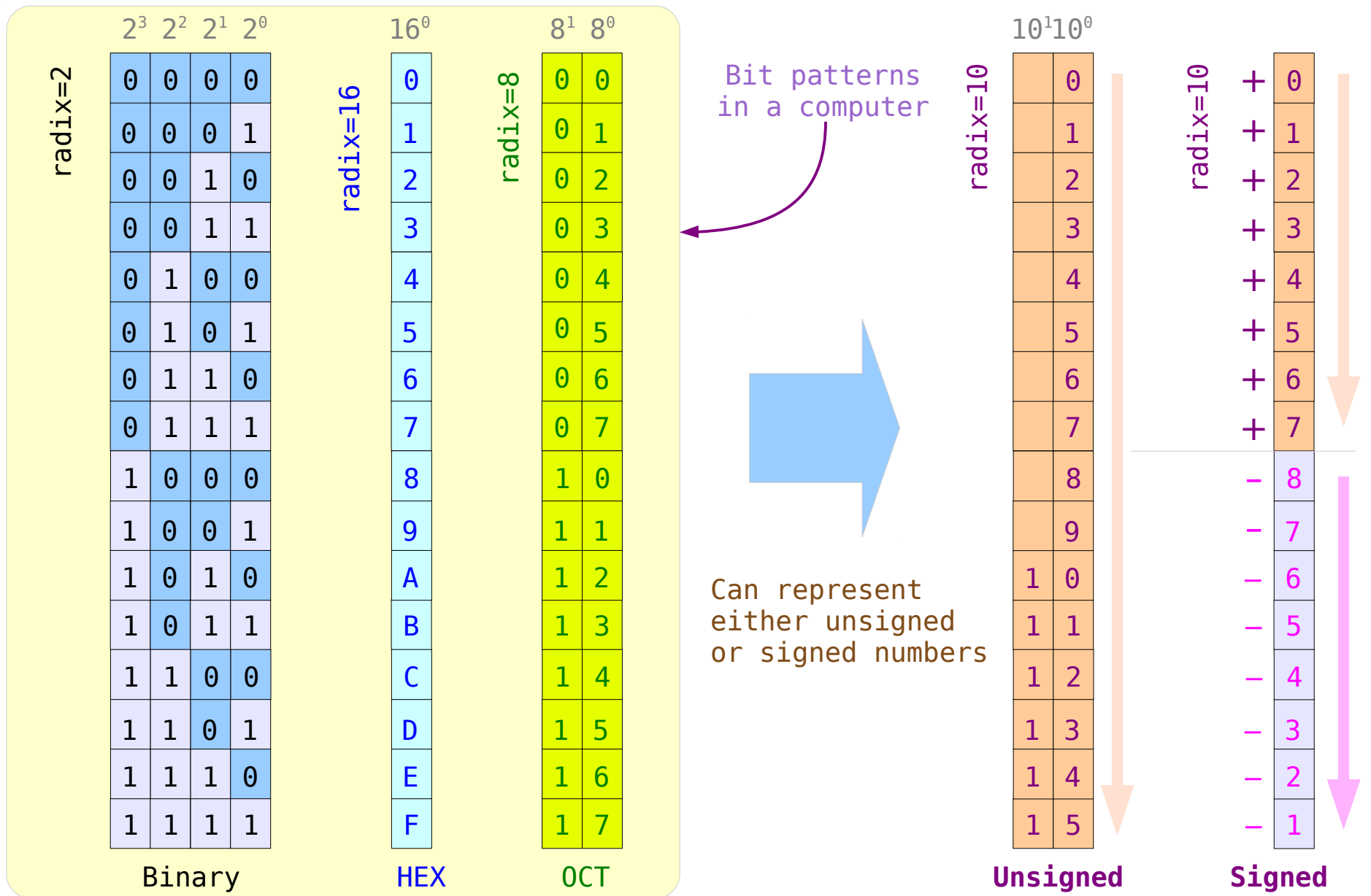
Bit patterns stored in a computer memory system



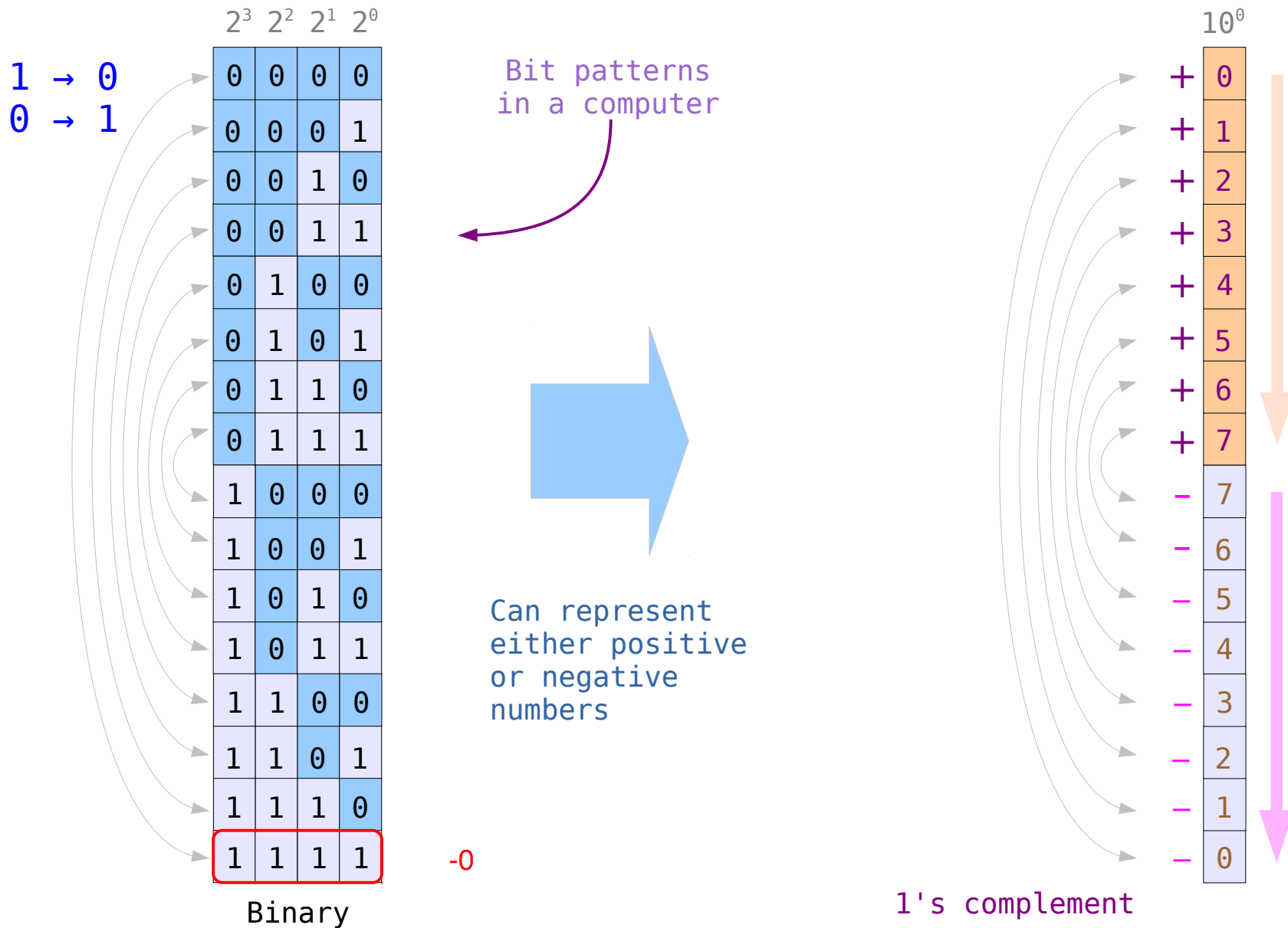
- Unsigned
- Signed

Each bit pattern can represent either unsigned or signed numbers

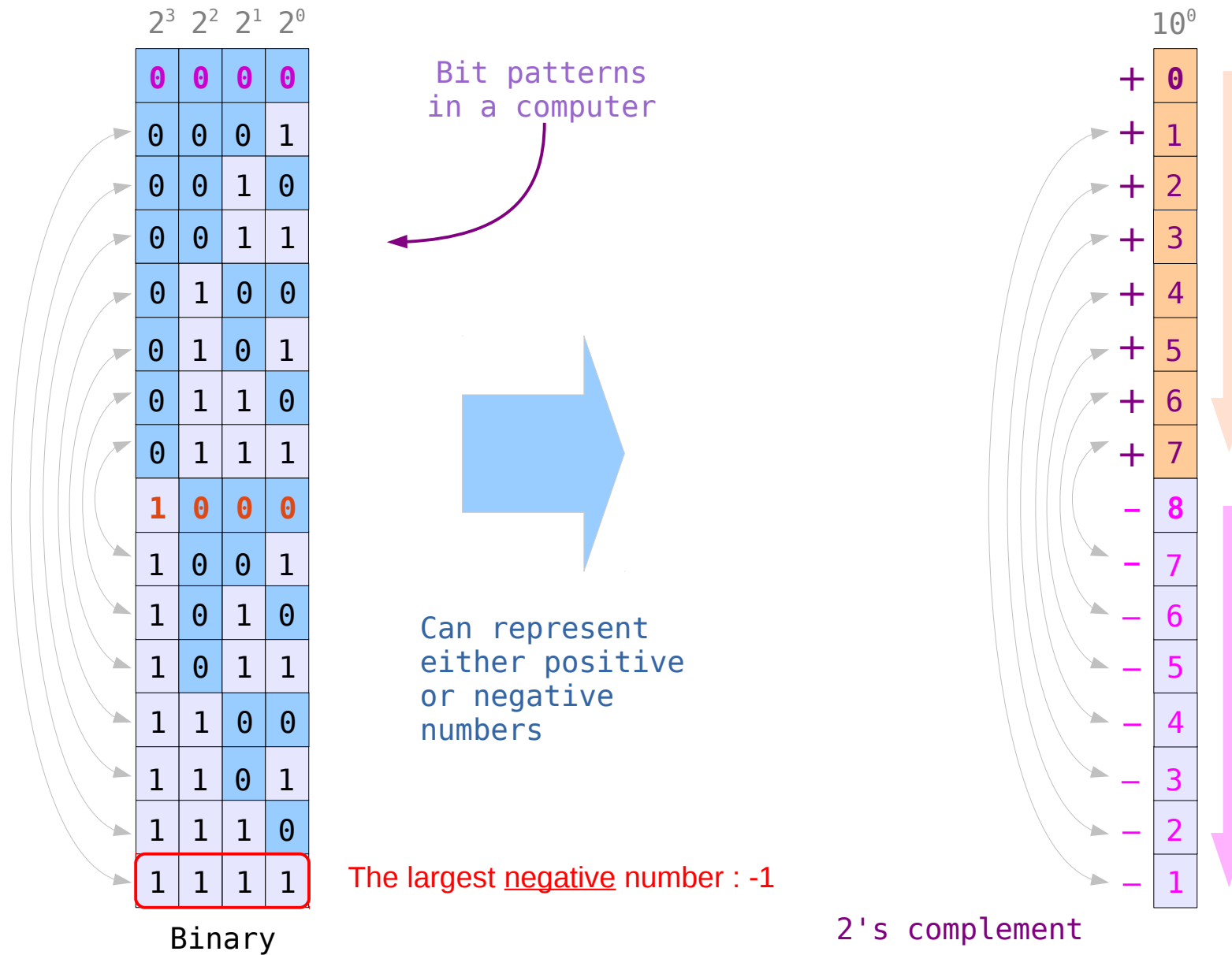
# Signed and Unsigned Numbers



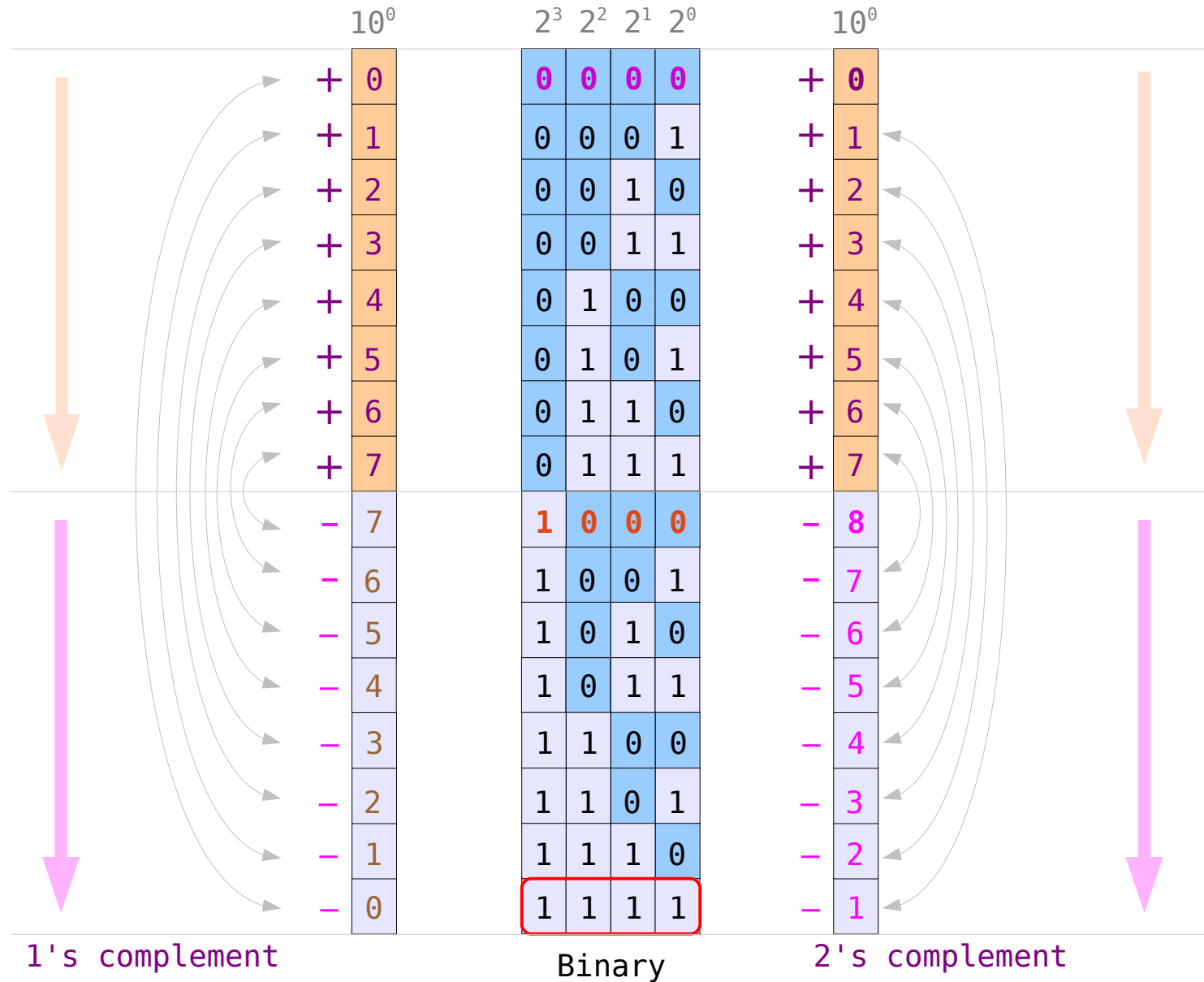
# 1's Complement signed numbers



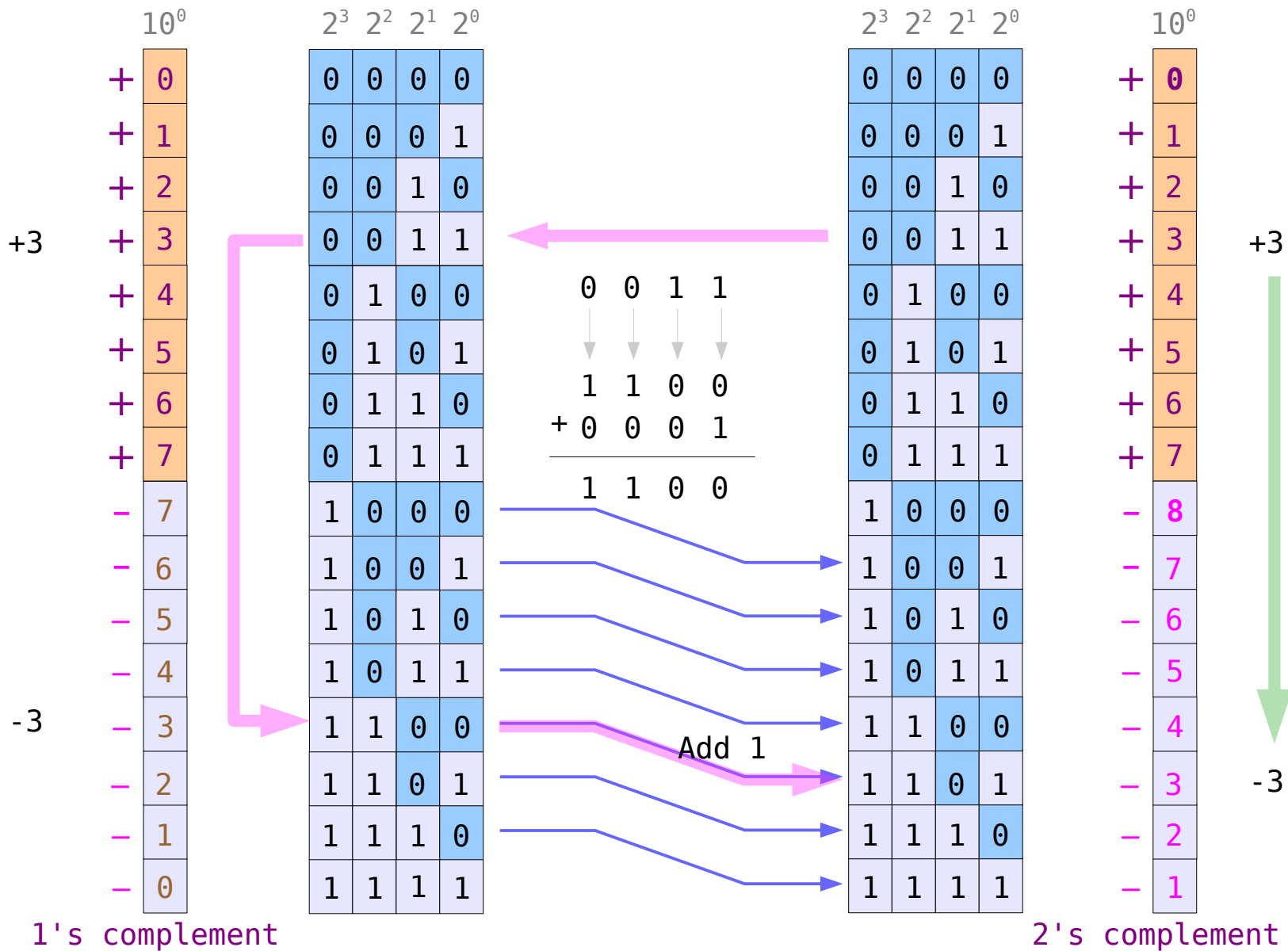
# 2's Complement signed numbers



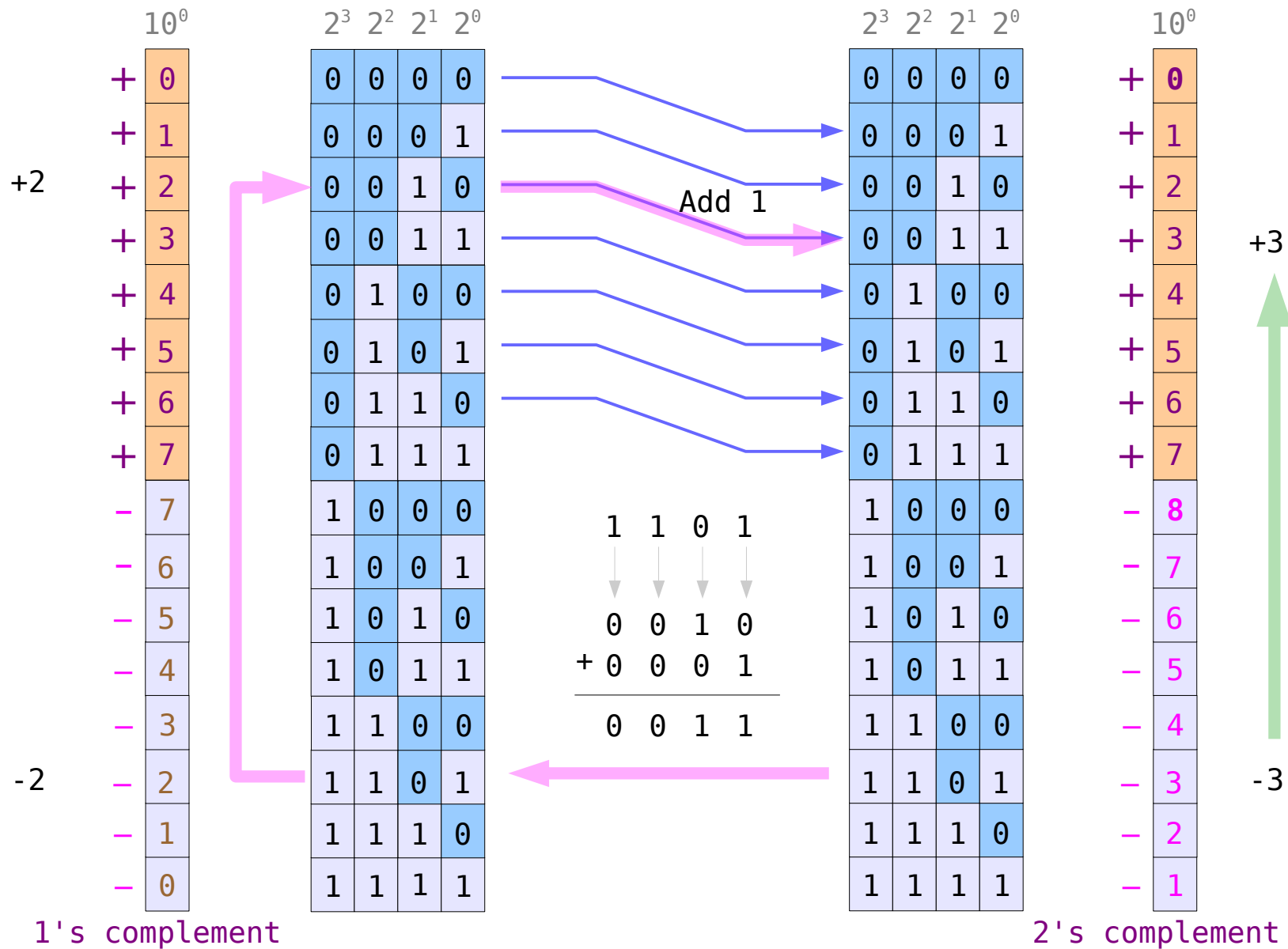
# 1's and 2's complement signed numbers



# Compute 2's Complement (+3 → -3)

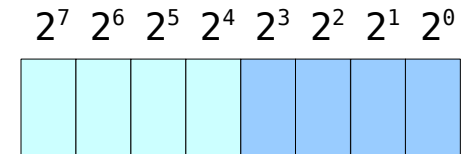


# Compute 2's Complement (-3 → +3)

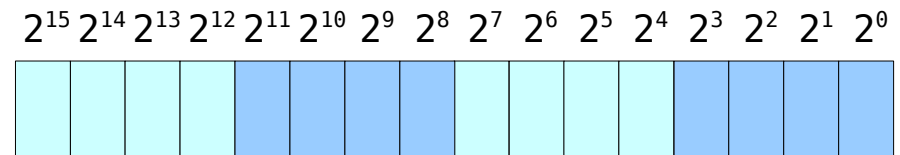


# Types of Integer Numbers

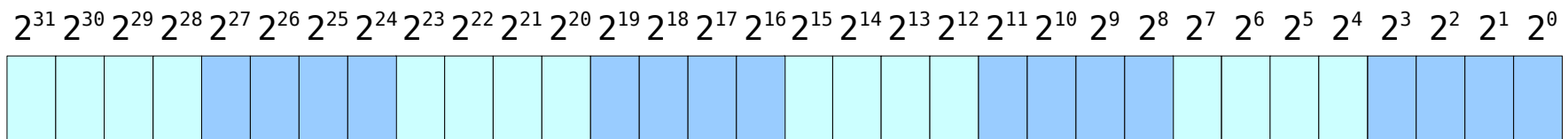
1 byte : **char**  $2^8$



2 bytes: **short**  $2^{16}$



4 bytes: **int**  $2^{32}$

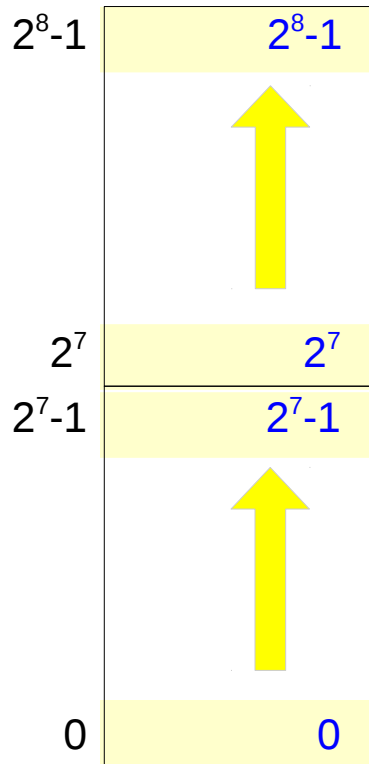




# Unsigned Integer Ranges

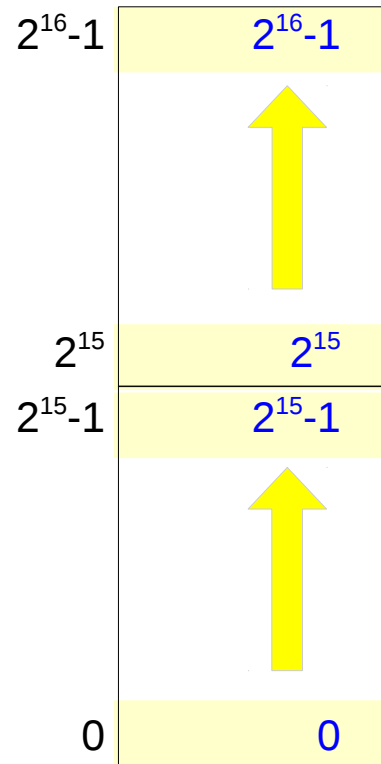
1 Byte

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
-------	-------	-------	-------	-------	-------	-------	-------



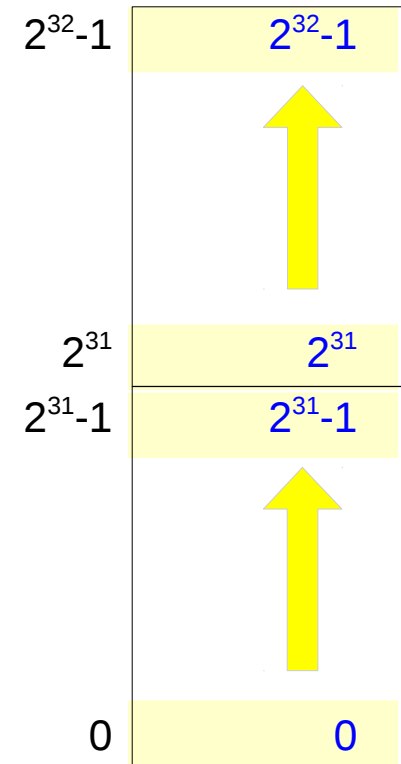
2 Bytes

$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$



4 Bytes

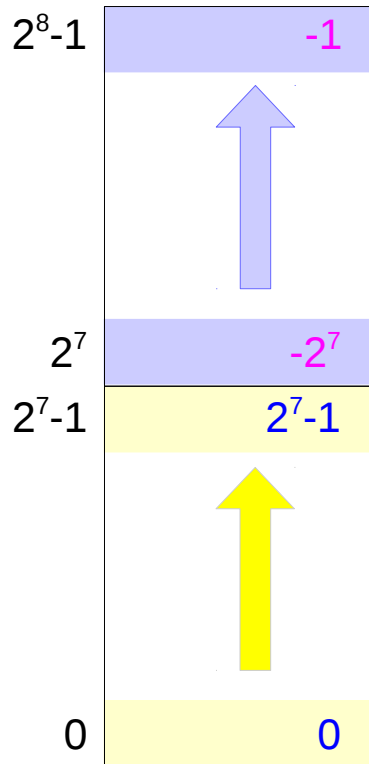
$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$
$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$



# Signed Integer Ranges

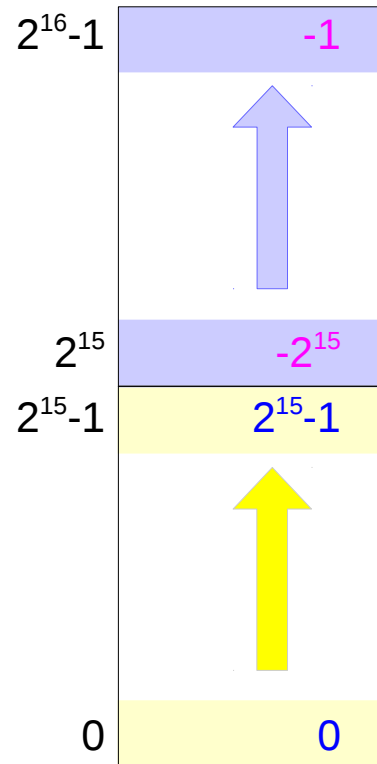
1 Byte

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
-------	-------	-------	-------	-------	-------	-------	-------



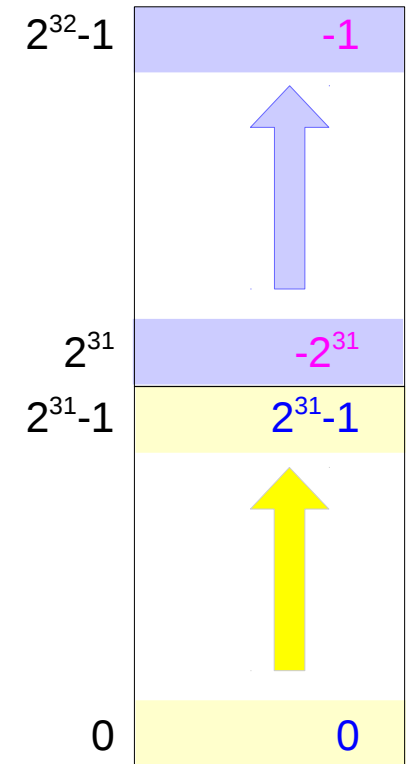
2 Bytes

$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$



4 Bytes

$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$
$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$



# Integer Ranges

1 Byte

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
-------	-------	-------	-------	-------	-------	-------	-------

unsigned char

$$[0, +(2^8-1)]$$

(signed) char

$$[0, +(2^7-1)]$$

$$[-2^7, -1]$$

2 Bytes

$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

unsigned short

$$[0, +(2^{16}-1)]$$

(signed) short

$$[0, +(2^{15}-1)]$$

$$[-2^{15}, -1]$$

4 Bytes

$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$
$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

unsigned int

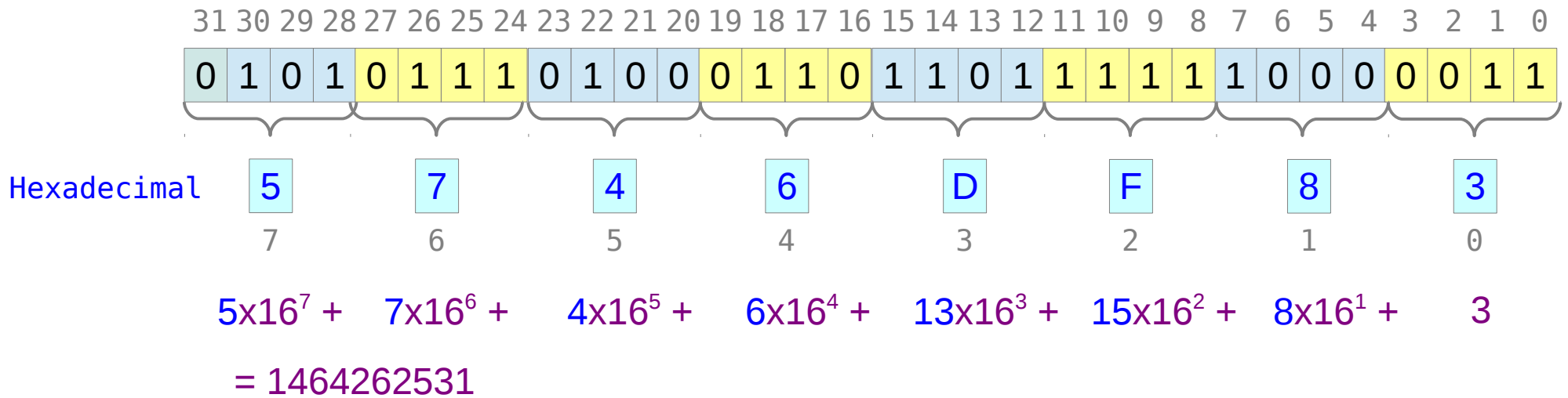
$$[0, +(2^{32}-1)]$$

(signed) int

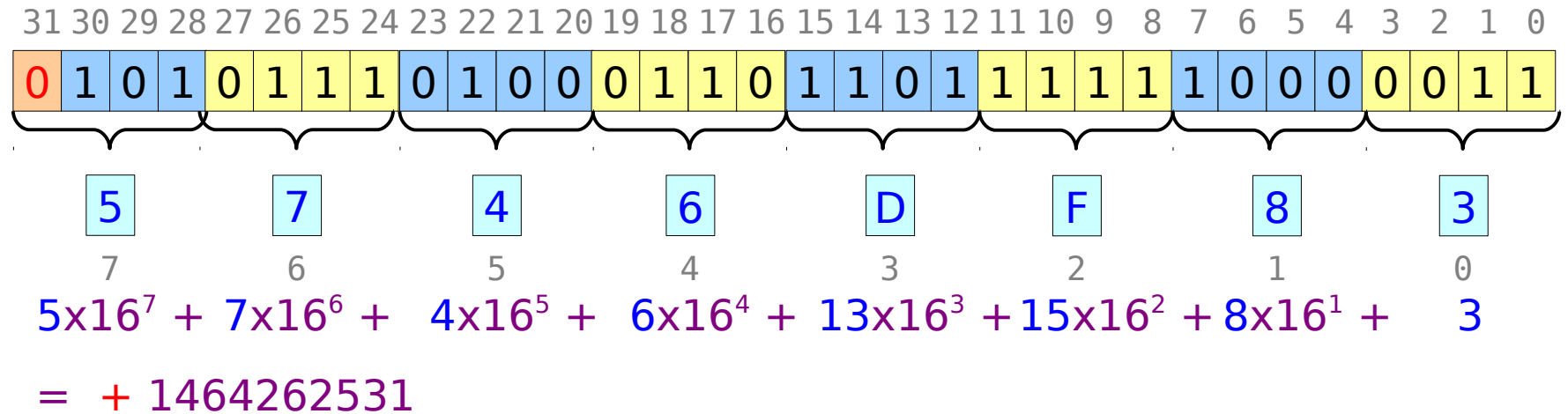
$$[0, +(2^{31}-1)]$$

$$[-2^{31}, -1]$$

# 4 Byte **Unsigned** Integer Example



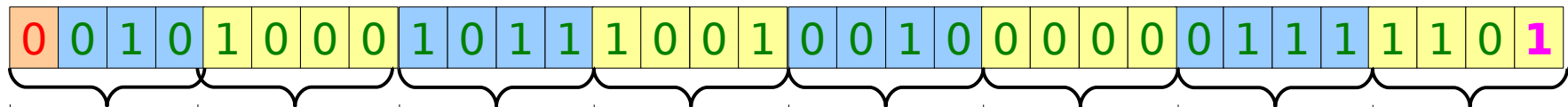
## 4 Byte **Signed** Integer - positive number



# 4 Byte **Signed** Integer - negative number



**2's complement**



$$-(2 \times 16^7 + 8 \times 16^6 + 11 \times 16^5 + 9 \times 16^4 + 2 \times 16^3 + 0 \times 16^2 + 7 \times 16^1 + 13)$$

$$= -683221117$$

# Sign Extension (1)

4-bit  
Signed  
Number

8-bit  
Signed  
Number

16-bit  
Signed  
Number

(+3)

(+3)

(+3)

3 2 1 0  
0 0 1 1

7 6 5 4 3 2 1 0  
0 0 0 0 0 0 1 1

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

(-3)

(-3)

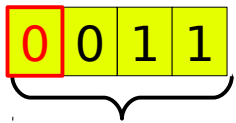
(-3)

3 2 1 0  
1 1 0 1

7 6 5 4 3 2 1 0  
1 1 1 1 1 1 0 1

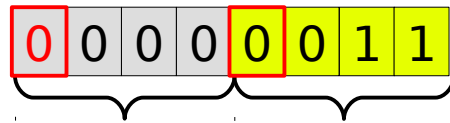
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1

# Sign Extension (2)



3

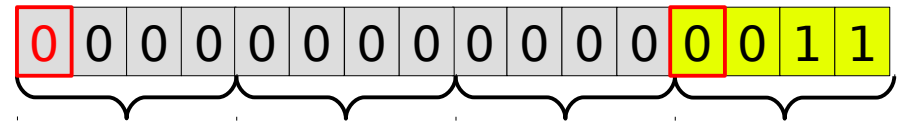
3



0

3

$0 \times 16^1 + 3$



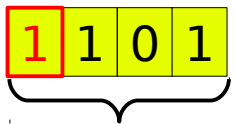
0

0

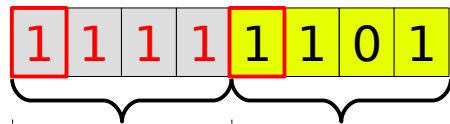
0

3

$0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 3$

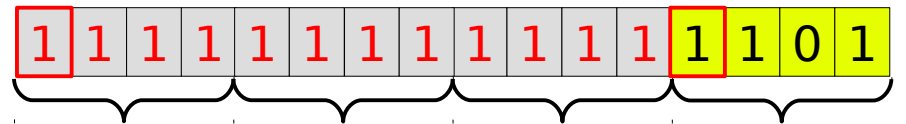


D



F

D

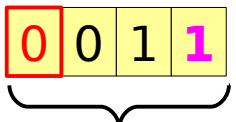


F

F

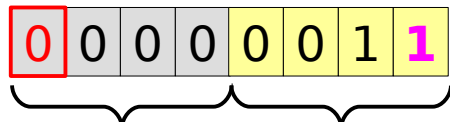
F

D



3

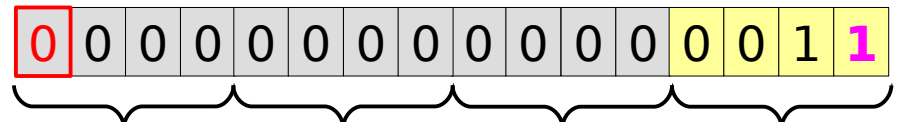
$-(3)$



0

3

$-(0 \times 16^1 + 3)$



0

0

0

3

$-(0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 3)$

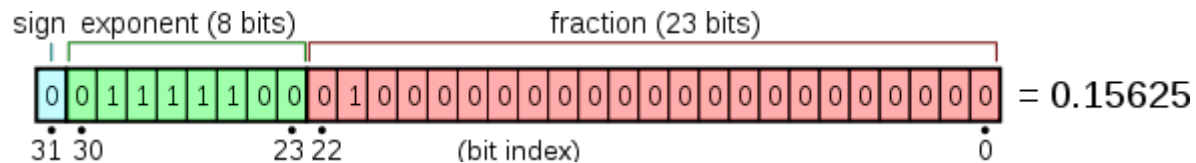


# Sign Extension (3)

				$16^0$					$2^3$	$2^2$	$2^1$	$2^0$
...	0	0	0	0	...	0	0	0	0	0	0	0
...	0	0	0	1	...	0	0	0	0	0	0	1
...	0	0	0	2	...	0	0	0	0	0	1	0
...	0	0	0	3	...	0	0	0	0	0	1	1
...	0	0	0	4	...	0	0	0	0	1	0	0
...	0	0	0	5	...	0	0	0	0	1	0	1
...	0	0	0	6	...	0	0	0	0	1	1	0
...	0	0	0	7	...	0	0	0	0	1	1	1
...	F	F	F	8	...	1	1	1	1	0	0	0
...	F	F	F	9	...	1	1	1	1	0	0	1
...	F	F	F	A	...	1	1	1	1	0	1	0
...	F	F	F	B	...	1	1	1	1	0	1	1
...	F	F	F	C	...	1	1	1	1	1	0	0
...	F	F	F	D	...	1	1	1	1	1	0	1
...	F	F	F	E	...	1	1	1	1	1	1	0
...	F	F	F	F	...	1	1	1	1	1	1	1

Hexadecimal Binary

# Floating Point Number Format

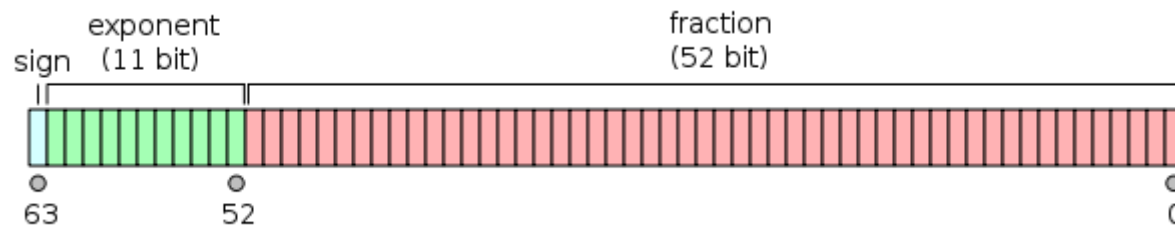


## float

- Sign (1-bit)
- Exponent (8-bits)
- Fraction (23-bits)

## 4 Bytes

$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$
$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$



## double

- Sign (1-bit)
- Exponent (11-bits)
- Fraction (52-bits)

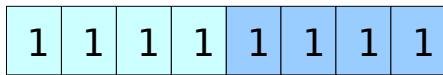
## 8 Bytes

$2^{63}$	$2^{62}$	$2^{61}$	$2^{60}$	$2^{59}$	$2^{58}$	$2^{57}$	$2^{56}$
$2^{55}$	$2^{54}$	$2^{53}$	$2^{52}$	$2^{51}$	$2^{50}$	$2^{49}$	$2^{48}$
$2^{47}$	$2^{46}$	$2^{45}$	$2^{44}$	$2^{43}$	$2^{42}$	$2^{41}$	$2^{40}$
$2^{39}$	$2^{38}$	$2^{37}$	$2^{36}$	$2^{35}$	$2^{34}$	$2^{33}$	$2^{32}$
$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$
$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

[https://en.wikipedia.org/wiki/Single-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Single-precision_floating-point_format)

# Signed and Unsigned Number Examples

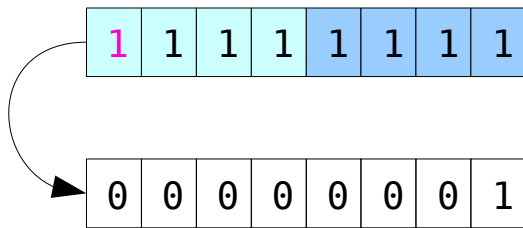
unsigned char u;



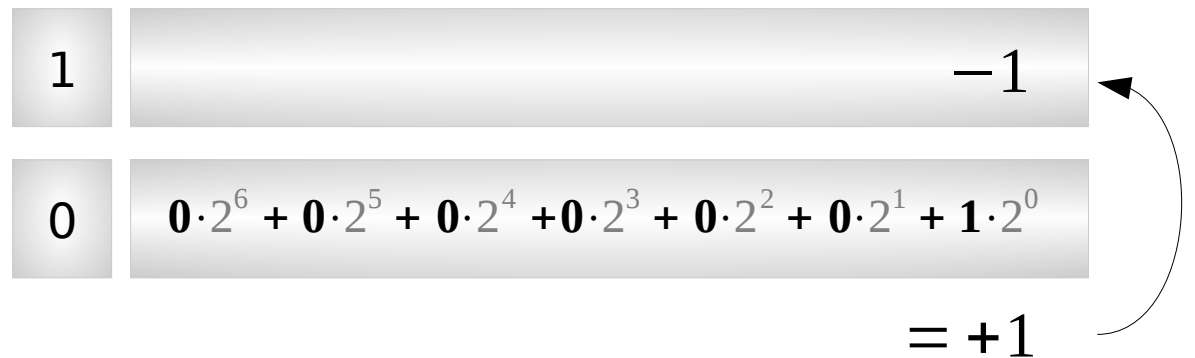
$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$2^8 - 1 = +255$$

(signed) char i;

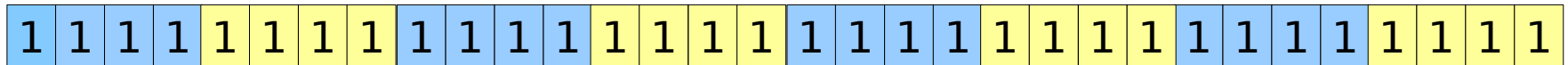


2's complement



# Conversion Specifier

`unsigned int u;`



$$2^{32} - 1$$



`%d` → **-1** (signed)  
`%u` → **4294967295** (unsigned)

`(signed) int i;`



`%d` → **-1** (signed)  
`%u` → **4294967295** (unsigned)

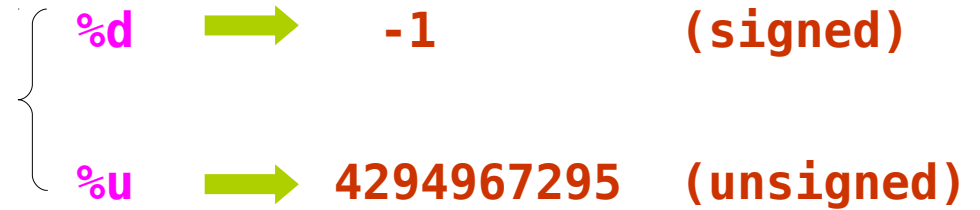
# Conversion Specifier

unsigned int u;



$$2^8 - 1 = +255$$

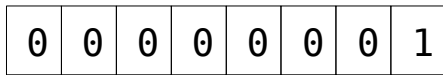
+255 →



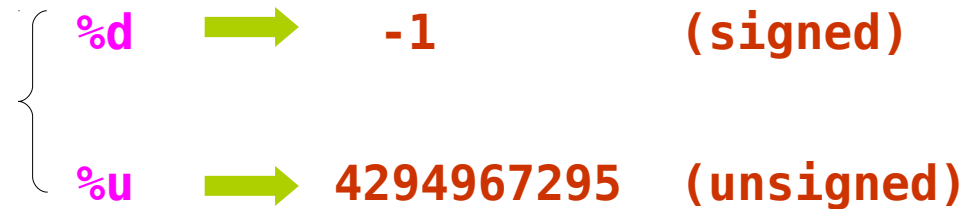
(signed) int i;



-1 →



+1



# Conversion Specifier

`unsigned char u;`

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

 +255

$$2^8 - 1 = +255$$

`(signed) char i;`

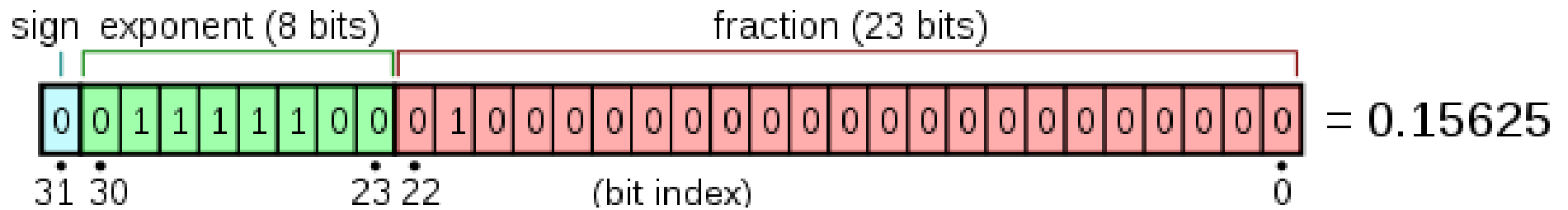
1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

 -1

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

 +1

# Single Precision : float



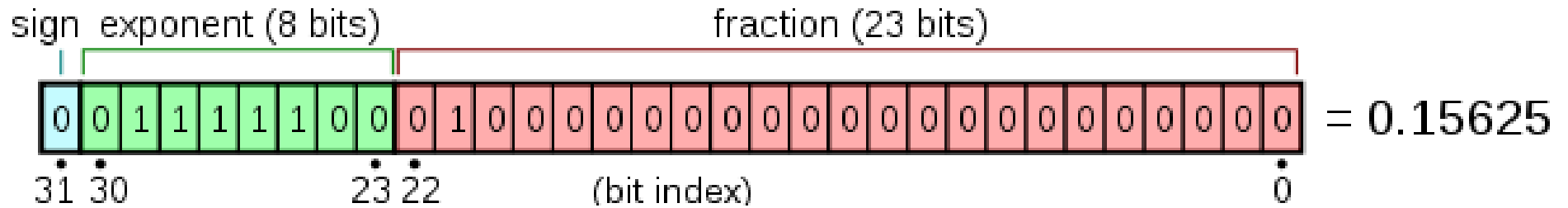
0: positive, 1: negative

exponent Excess +127 (must be subtracted)  $127 = 2^{8-1} - 1$



[https://en.wikipedia.org/wiki/Single-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Single-precision_floating-point_format)

# Single Precision : float



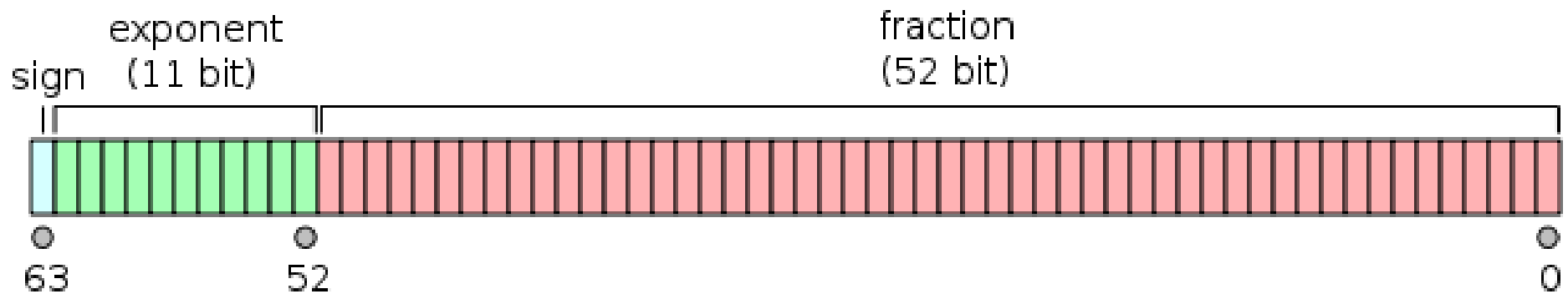
$$(-1)^{b_{31}} \times (1.b_{22}b_{21} \dots b_0)_2 \times 2^{(b_{30}b_{29} \dots b_{23})_2 - 127},$$

$$\text{value} = (-1)^{\text{sign}} \times \left( 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right) \times 2^{(e-127)}.$$

[https://en.wikipedia.org/wiki/Single-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Single-precision_floating-point_format)



# Double Precision : double



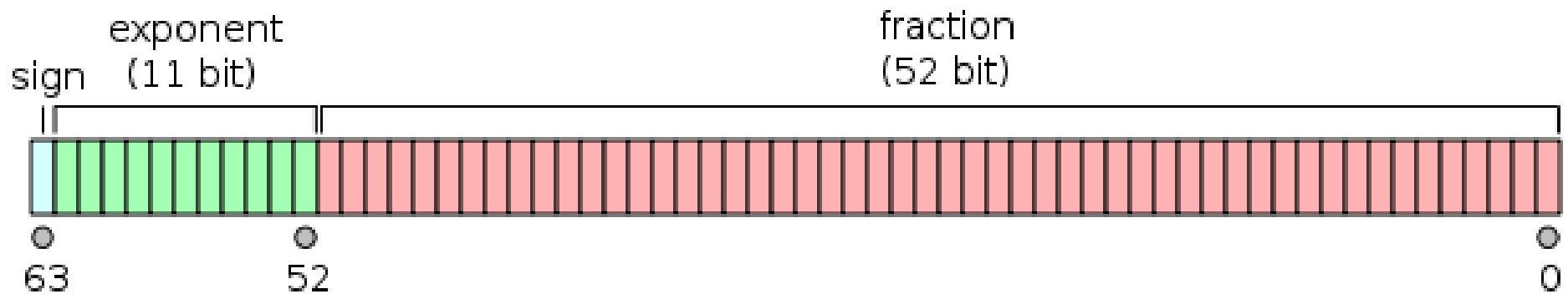
0: positive, 1: negative

**exponent** Excess +1023 (must be subtracted)  $1023 = 2^{11-1} - 1$

Implicit **1** fraction

[https://en.wikipedia.org/wiki/Double-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Double-precision_floating-point_format)

# Double Precision : double



$$(-1)^{\text{sign}} (1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}} \left( 1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \times 2^{e-1023}$$

[https://en.wikipedia.org/wiki/Double-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Double-precision_floating-point_format)

## References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun