Computers
Data Representation
Exercises

Grau en Ciència i Enginyeria de Dades

Facultat d'Informàtica de Barcelona (FIB)

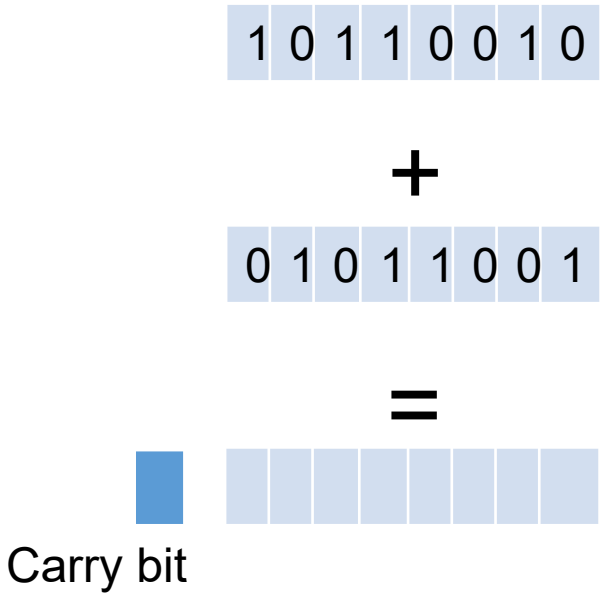
Universitat Politècnica de Catalunya (UPC)

2021-2022 Q2

Creative Commons License

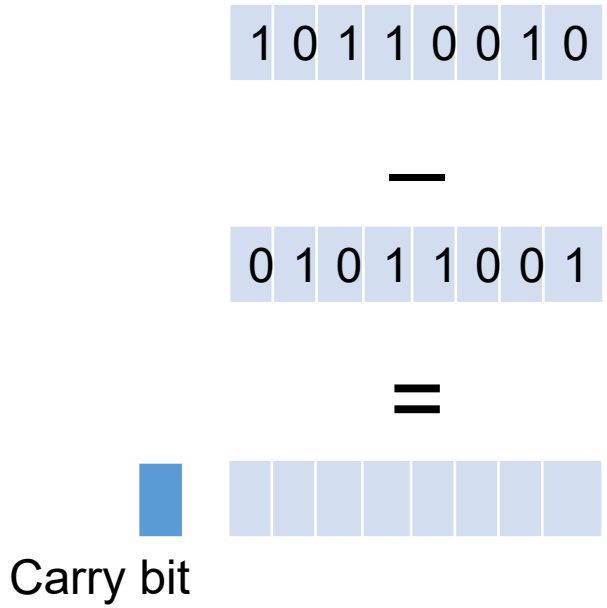
Binary arithmetic

- Binary addition



Binary arithmetic

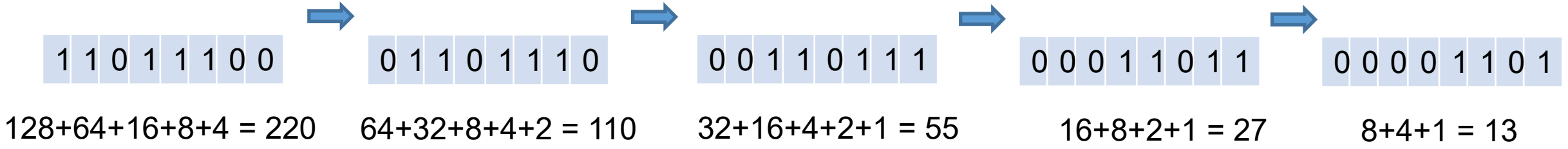
- Substraction



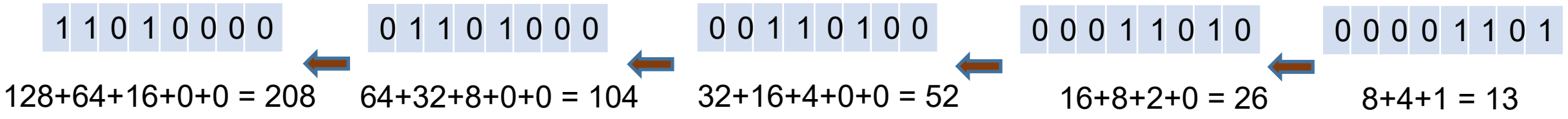
- do you see a property shared between the two numbers?

Binary arithmetic

- Division /2 - shift right
  - Right-most bit is lost

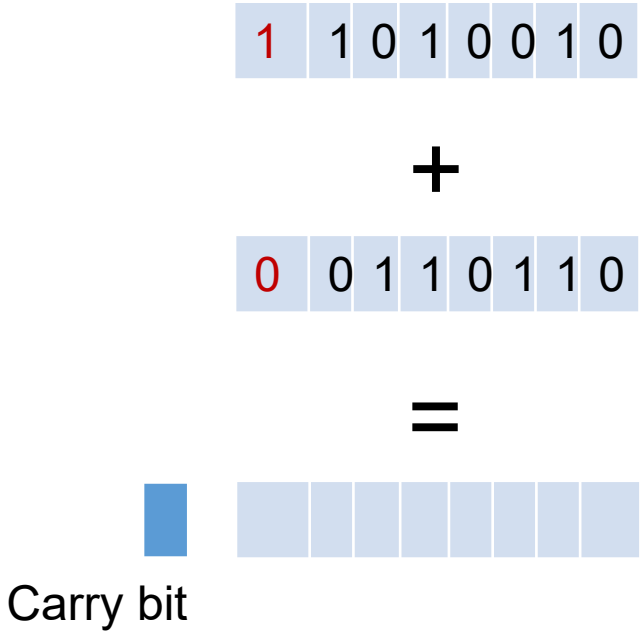


- Multiplication*2 - shift left
  - Left-most bit gets into the carry



Binary arithmetic

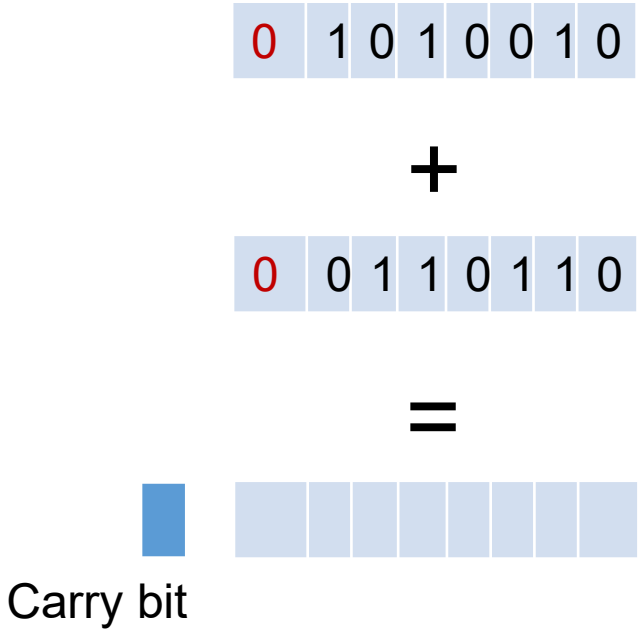
- Signed numbers ... most significant bit represents the sign



Into sign	Into carry	Overflow
0	0	0
0	1	1
1	0	1
1	1	0

Binary arithmetic

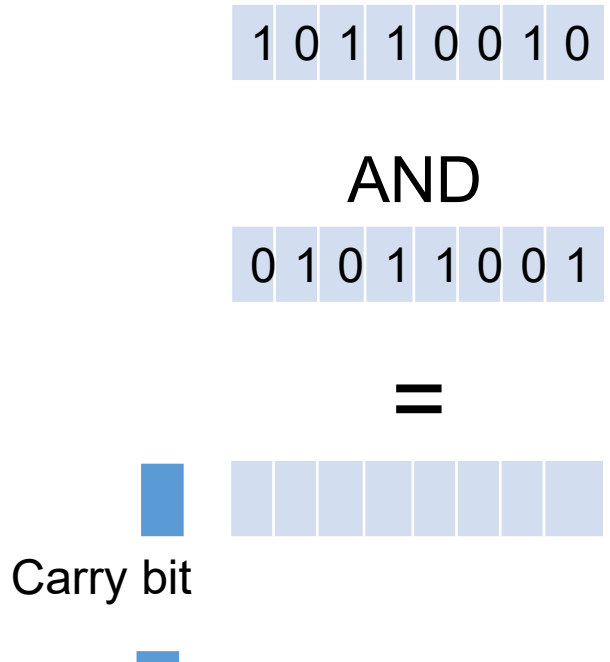
- Signed numbers ... most significant bit represents the sign



Into sign	Into carry	Overflow
0	0	0
0	1	1
1	0	1
1	1	0

Binary logical operations

- And

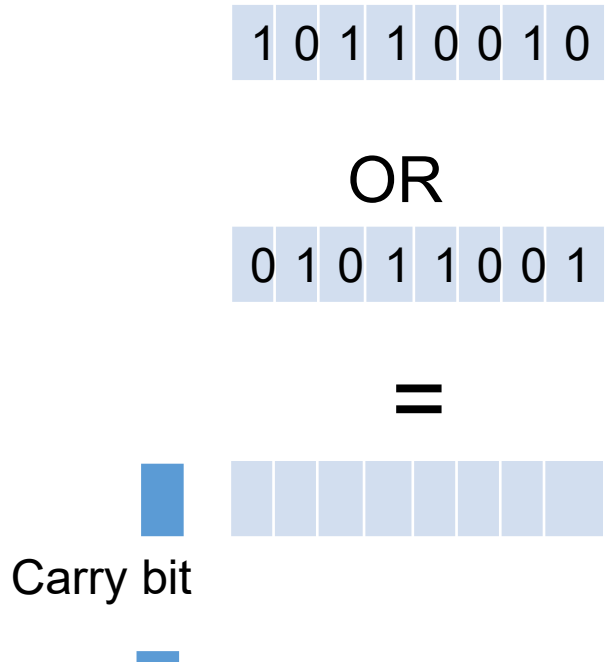


Src 0	Src 1	AND
0	0	0
0	1	0
1	0	0
1	1	1

*this operation may or may not modify the carry and overflow bits depending on the particular processor brand (Intel, ARM, IBM...)

Binary logical operations

- Or

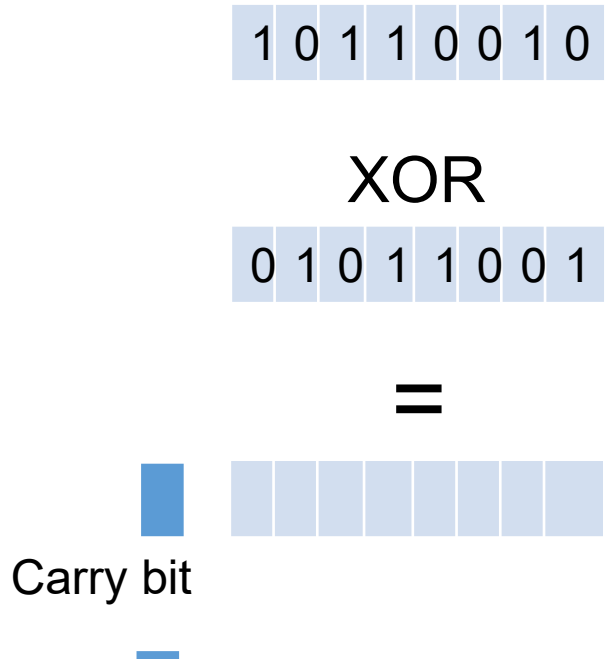


Src 0	Src 1	OR
0	0	0
0	1	1
1	0	1
1	1	1

*this operation may or may not modify the carry and overflow bits depending on the particular processor brand (Intel, ARM, IBM...)

Binary logical operations

- Xor – exclusive or

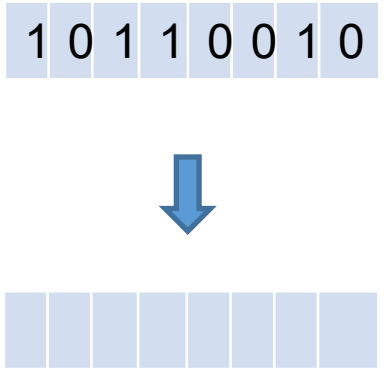


Src 0	Src 1	XOR
0	0	0
0	1	1
1	0	1
1	1	0

*this operation may or may not modify the carry and overflow bits depending on the particular processor brand (Intel, ARM, IBM...)

Unary logical operation

- Not



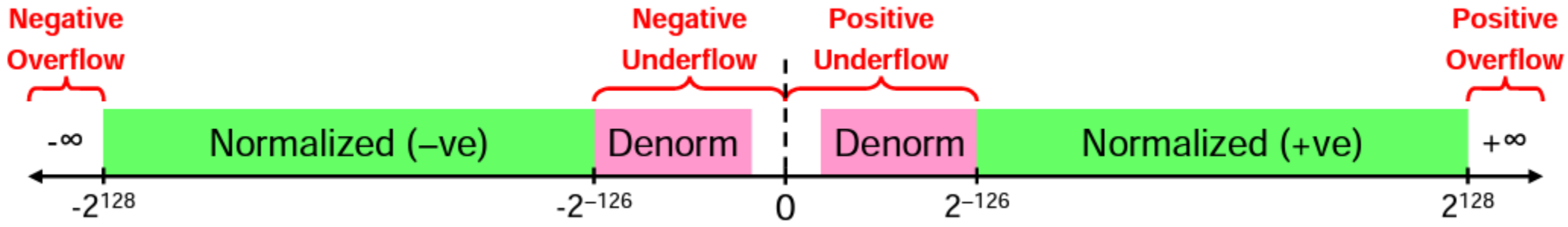
Src	NOT
0	1
1	0

*this operation may or may not modify the carry and overflow bits depending on the particular processor brand (Intel, ARM, IBM...)

Floating point

- Underflow vs Overflow
- Normalized vs Denormalized

Single-Precision	Exponent = 8	Fraction = 23	Value
Normalized Number	1 to 254	Anything	$\pm (1.F)_2 \times 2^{E-127}$
Denormalized Number	0	nonzero	$\pm (0.F)_2 \times 2^{-126}$
Zero	0	0	± 0
Infinity	255	0	$\pm \infty$
NaN	255	nonzero	NaN



Floating point

- $19{,}59375_{10}$
  - 1st Step: determine sign bit
  - 2nd Step: convert to binary
    - 19 →
    - 0,59375 →
  - 3rd Step: normalise the mantissa and determine the unbiased exponent
  - 4th Step: determine the biased exponent
  - 5th Step: remove the leading 1 from the mantissa

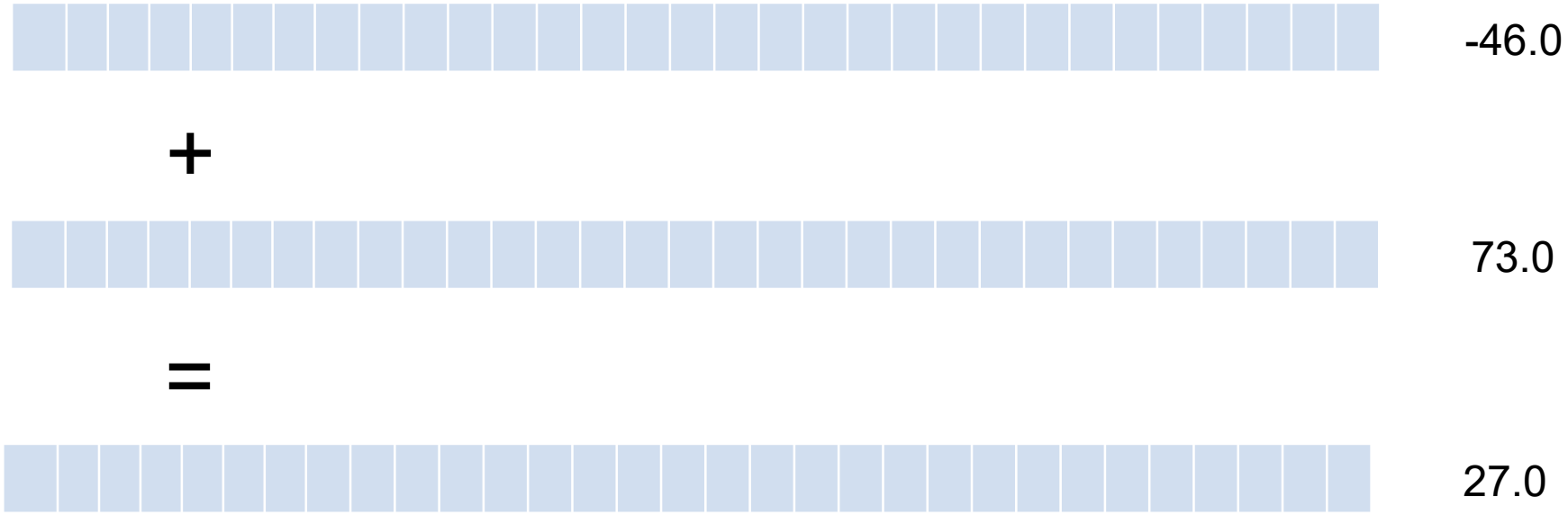
- Sol: **0 10000011 00111001110000000000000**

Floating point

- $0{,}09375_{10}$
  - Sol: **0 01111011 10000000000000000000000**
- $-123{,}3_{10}$
  - Sol: **1 10000101 11101101001100110011001**
    Sol: **1 10000101 11101101001100110011010**
- $0\ 10000100\ 11010100000000000000000_2$
  - Sol: **58,5**
- $1\ 00001001\ 00011001000000000000000_2$
  - Sol: **$-3{,}303139125810627897392149669594 5 * 10^{-36}$**

Floating point

- Sign + exponent + mantissa



Binary vs Logical Operations

Expression	Binary value	Hex value
a	0101 0101	
b	0100 0110	
a & b		
a | b		
a ^ b		
~a | ~b		
a & !b		
a && b		
a || b		
!a || !b		
a && ~b		