These slides are being provided with permission from the copyright for in-class (CS2208B) use only. The slides must not be reproduced or provided to anyone outside of the class.

All download copies of the slides and/or lecture recordings are for personal use only. Students must destroy these copies within 30 days after receipt of final course evaluations.

# Tutorial 03: Addition/Subtraction using 2's Complement

Computer Science Department

CS2208: Introduction to Computer Organization and Architecture

Winter 2020-2021

Instructor: Mahmoud R. El-Sakka

Office: MC-419

Email: elsakka@csd.uwo.ca

Phone: 519-661-2111 x86996



# **Binary Arithmetic**

☐ These tables cover the fundamental arithmetic operations.

Addition	Subtraction	Multiplication
0 + 0 = 0 (carry 0)	0 - 0 = 0 (borrow 0)	$0 \times 0 = 0$
0 + 1 = 1 (carry 0)	0 - 1 = 1 (borrow 1)	$0 \times 1 = 0$
1 + 0 = 1 (carry 0)	1 - 0 = 1 (borrow 0)	$1 \times 0 = 0$
1 + 1 = 0 (carry 1)	1 - 1 = 0 (borrow 0)	$1 \times 1 = 1$

#### **Addition (three bits)**

#### **Subtraction (three bits)**

1 - 1 - 1 = 1 (borrow 1)

# Sign and Magnitude Addition/Subtraction

- The operations are carried out similar to normal math calculations
- The resultant sign is arranged separately
  - $\square$  The sign of A B depends on the values of A and B
  - $\square$  If B > A, the answer will be calculated as -(B A), O.W., it is +(A B)
- The location of the radix points needs to be aligned before performing the operation.
- If the provided number of bits are not enough to hold the result, it means an overflow occurred.

- A subtraction operation is converted to an addition operation (after performing the 2's complement to the operand appearing after the negative sign)
- When adding two *positive* numbers and finding the result is *negative*, this means an *overflow occurred*.
- When adding two *negative* numbers and finding the result is *positive*, this means an *overflow occurred*.
- Overflow will never occur when adding a positive number to a negative number, or vice versa.
- How about
  - □ subtracting a negative number from a positive number?
  - □ subtracting a positive number from a negative number?

**■** *Example 1*:

Perform  $20_{10} - 10_{10}$  using 2's complement 6-bit system

- $20_{10} \rightarrow 10100_2$
- $10_{10} \rightarrow 1010_{2}$
- $20_{10} 10_{10} \rightarrow 10100_2 1010_2$ 
  - $\rightarrow$  010100<sub>2</sub> 001010<sub>2</sub>
  - $\rightarrow 010100_2 + (-001010_2)$
- This is the answer in 2's complement

This is the

- **→** 010100<sub>2</sub> +
- $^{\circ}$   $\rightarrow$  001010<sub>2</sub>
- answer in  $\rightarrow +10_{10}$ decimal to verify

to be ignored +110110 110110 1001010, **Overflow** can not occur

010100,

**Carry out** 

**■** *Example 2*:

Perform  $10_{10} - 20_{10}$  using 2's complement 6-bit system

- $\bullet$  10<sub>10</sub>  $\rightarrow$  1010<sub>2</sub>
- $\bullet$  20<sub>10</sub>  $\rightarrow$  10100<sub>2</sub>
- $\blacksquare 10_{10} 20_{10} \rightarrow 1010_2 10100_2$ 
  - $\rightarrow$  001010<sub>2</sub> 010100<sub>2</sub>
  - $\rightarrow$  001010<sub>2</sub> + (-010100<sub>2</sub>)
- This is the answer in 2's complement

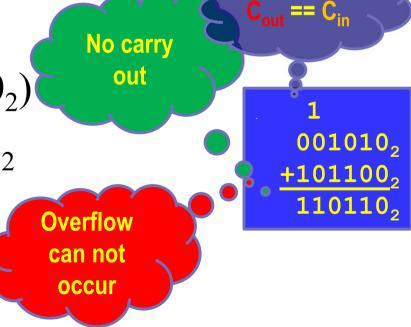
This is the

answer in

decimal to verify

- $\rightarrow$  001010<sub>2</sub> + 101100<sub>2</sub>
- **→** 110110<sub>2</sub>
- **→**-001010<sub>2</sub>

**→**-10<sub>10</sub>



© Mahmoud R. El-Sakka

**■** *Example 3*:

Perform  $20_{10} + 10_{10}$  using 2's complement 6-bit system

- $20_{10} \rightarrow 10100_2$
- $10_{10}$  →  $1010_2$

$$\blacksquare 20_{10} + 10_{10} \rightarrow 10100_2 + 1010_2$$

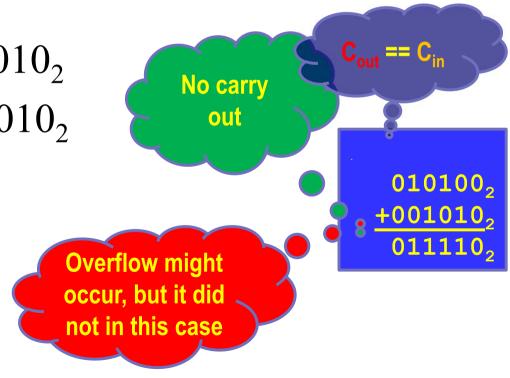
This is the answer in 2's complement

 $\rightarrow$  010100<sub>2</sub> + 001010<sub>2</sub>

 $^{\circ}$   $\rightarrow$  0111110<sub>2</sub>

This is the answer in decimal to verify

$$\rightarrow +30_{10}$$



**■** *Example 4*:

Perform  $-20_{10} - 10_{10}$  using 2's complement 6-bit system

- $\bullet$  20<sub>10</sub>  $\rightarrow$  10100<sub>2</sub>
- $\blacksquare 10_{10} \rightarrow 1010_2$
- $-20_{10} 10_{10} \rightarrow -10100_2 1010_2$ 
  - $\rightarrow$  -010100<sub>2</sub> 001010<sub>2</sub>
  - $\rightarrow$  (-010100<sub>2</sub>)+ (-001010<sub>2</sub>)

This is the answer in 2's complement

This is the

 $\rightarrow$  101100<sub>2</sub> + 110110<sub>2</sub>

8

- $\rightarrow$  100010<sub>2</sub>
- **→** -011110<sub>2</sub>

answer in decimal to verify  $\rightarrow$   $-30_{10}$ 

Carry out to be ignored

Overflow might occur, but it did not in this case

1111 101100<sub>2</sub> +110110<sub>2</sub> 1100010<sub>2</sub>

CS 2208: Introduction to Computer Organization and Architecture

**■** *Example 5*:

Perform  $20_{10} + 20_{10}$  using 2's complement 6-bit system

 $\bullet$  20<sub>10</sub>  $\rightarrow$  10100<sub>2</sub>

■ 
$$20_{10} + 20_{10}$$
 →  $10100_2 + 10100_2$   
→  $010100_2 + 010100_2$   
No carry out

Overflow might occur, and indeed it did in this case

- **■** *Example 6*:
  - Perform  $-20_{10} 20_{10}$  using 2's complement 6-bit system
- $20_{10} \rightarrow 10100_2$
- $-20_{10} 20_{10} \rightarrow -10100_2 10100_2$ 
  - $\rightarrow$  -010100<sub>2</sub> 010100<sub>2</sub>
  - $\rightarrow$  (-010100<sub>2</sub>)+ (-010100<sub>2</sub>) Carry out
  - $\rightarrow$  101100<sub>2</sub> + 101100<sub>2</sub>

Carry out to be ignored

Overflow might occur, and indeed it did in this case

1 11 101100<sub>2</sub> +101100<sub>2</sub> 1011000<sub>2</sub>

**■** *Example 7*:

Perform  $20_{10} - 20_{10}$  using 2's complement 6-bit system

- $\blacksquare 20_{10} \rightarrow 10100_2$
- $\blacksquare 20_{10} 20_{10} \rightarrow 10100_2 10100_2$ 
  - $\rightarrow$  010100<sub>2</sub> 010100<sub>2</sub>
  - $\rightarrow$  010100<sub>2</sub> + (-010100<sub>2</sub>)
  - $\rightarrow$  010100<sub>2</sub> + 101100<sub>2</sub>
  - $\rightarrow$  000000<sub>2</sub>
- answer in decimal to verify  $\longrightarrow 0_1$



This is the answer

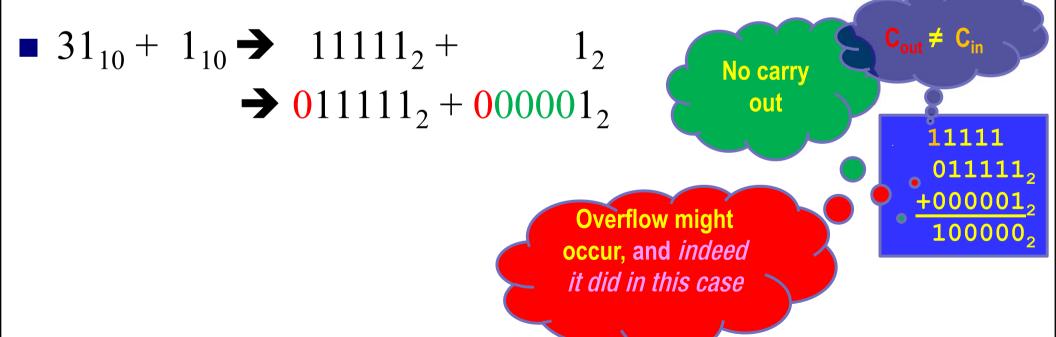
in 2's complement

This is the

**■** *Example 8*:

Perform  $31_{10} + 1_{10}$  using 2's complement 6-bit system

- $\blacksquare$  31<sub>10</sub>  $\rightarrow$  11111<sub>2</sub>



**■** *Example 9*:

Perform  $-31_{10}$  –  $1_{10}$  using 2's complement 6-bit system

- $\blacksquare$  31<sub>10</sub>  $\rightarrow$  11111<sub>2</sub>

Carry out to be ignored

$$-31_{10} - 1_{10} \rightarrow -111111_2 -$$

$$\rightarrow$$
 (-0111111<sub>2</sub>) + (-000001<sub>2</sub>)

This is the answer in 2's complement

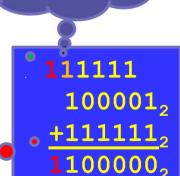
This is the

$$\rightarrow$$
 ( 100001<sub>2</sub>) + ( 111111<sub>2</sub>)

- $\rightarrow$  100000<sub>2</sub>
- $\rightarrow$  -100000<sub>2</sub>

answer in decimal to verify 
$$\rightarrow$$
  $-32_{10}$ 

Overflow might occur, but it did not in this case



© Mahmoud R. El-Sakka

**■** *Example 10*:

Encode –3.25<sub>10</sub> using 2's complement 6-bit system

- $\blacksquare$  3.25<sub>10</sub>  $\rightarrow$  11.01<sub>2</sub>
- $-3.25_{10} \rightarrow -0011.01_2$ 
  - **→** 1100.11<sub>2</sub>

Carry out to be ignored

You can also look at it as if it is  $-3_{10}$   $-0.25_{10}$ 

- $-3_{10} 0.25_{10} \rightarrow -11_2 0.01_2$ 
  - $\rightarrow (-000011_2) + (-0000.01_2)$

This is the answer in 2's complement

- $\rightarrow$  ( 111101<sub>2</sub>) + ( 1111.11<sub>2</sub>)
- **→** 111100.11<sub>2</sub>



1100.11<sub>2</sub>

 $-3.25_{10}$ 

Overflow might occur, but it did not in this case

Binary points
MUST be
aligned

 $C_{out} == C_{in}$ 

111111

111101.00

•+111111.11,

1111100.11,

puter Organization and Architecture

© Mahmoud R. El-Sakka

14 CS 2208: Intro