These slides are being provided with permission from the copyright for CS2208 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

Fall 2022-2023

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)

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

- \bullet 20₁₀ \rightarrow 10100₂
- 10_{10} → 1010_2

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

- \rightarrow 010100₂ 001010₂
- \rightarrow 010100₂ + (-001010₂)

This is the answer in 2's complement

- → 010100₂ +
- \rightarrow 001010₂

This step is not needed. It is just for you to verify.
$$\longrightarrow$$
 +10₁₀

110110₂
Overflow can not occur

11 1 010100₂ +110110₂ 1001010₂

© Mahmoud R. El-Sakka

Carry out

to be

ignored

■ *Example 2*:

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

- $\blacksquare 10_{10} \rightarrow 1010_2$
- \bullet 20₁₀ \rightarrow 10100₂

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

$$\rightarrow$$
 001010₂ - 010100₂

$$\rightarrow$$
 001010₂ + (-010100₂)

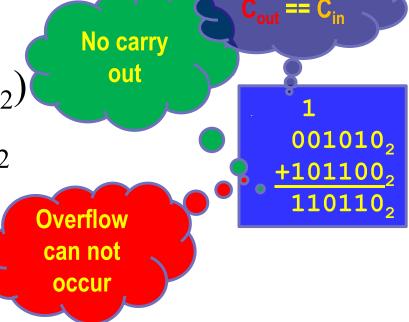
This is the answer in 2's complement

$$\rightarrow$$
 001010₂ + 101100₂

$$\rightarrow$$
 110110₂

$$-001010_2$$

This step is not needed. It is just for you to verify.



■ *Example 3*:

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

- \bullet 20₁₀ \rightarrow 10100₂
- $\blacksquare 10_{10} \rightarrow 1010_2$

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

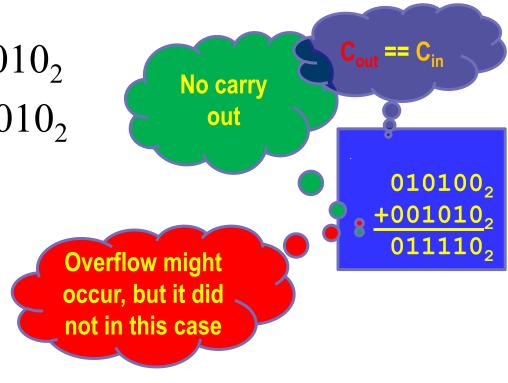
This is the answer in 2's complement

 \rightarrow 010100₂ + 001010₂

 $^{\circ}$ \rightarrow 0111110₂

This step is not needed. It is just for you to verify.

 $\rightarrow +30_{10}$



Example 4:

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

- $20_{10} \rightarrow 10100_2$
- $10_{10} \rightarrow 1010_{2}$

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

$$\rightarrow$$
 -010100₂ - 001010₂

$$\rightarrow$$
 (-010100₂)+ (-001010₂)

This is the answer in 2's complement

$$\rightarrow$$
 101100₂+

8

$$-011110_2$$

$$\rightarrow$$
 -30₁₀

Overflow might occur, but it did not in this case

110110₂

Carry out to be ignored

1111

101100

1100010_a

needed. It is just, for vou to verify

This step is not

© Mahmoud R. El-Sakka

CS 2208: Introduction to Computer Organization and Architecture

■ *Example 5*:

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

 \bullet 20₁₀ \rightarrow 10100₂

■
$$20_{10} + 20_{10}$$
 → $10100_2 + 10100_2$
→ $010100_2 + 010100_2$

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
- \bullet 20₁₀ \rightarrow 10100₂
- $-20_{10} 20_{10} \rightarrow -10100_2 10100_2$
 - \rightarrow -010100₂ 010100₂
 - \rightarrow (-010100₂)+ (-010100₂) Carry out
 - \rightarrow 101100₂ + 101100₂

Carry out to be ignored

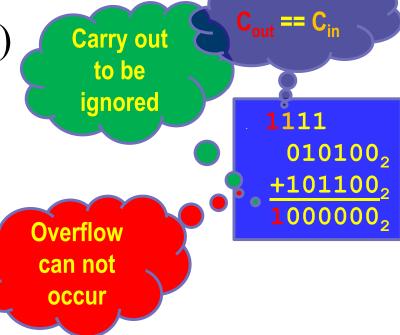
Overflow might occur, and indeed it did in this case

1 11 101100₂ +101100₂ 1011000₃

■ *Example 7*:

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

- $\blacksquare 20_{10} \rightarrow 10100_{2}$
- $20_{10} 20_{10} \rightarrow 10100_{2} 10100_{2}$
 - \rightarrow 010100₂ 010100₂
 - \rightarrow 010100₂ + (-010100₂)
- $\rightarrow 010100_2 + 101100_2$ in 2's complement
 - \rightarrow 000000₂
- needed. It is just, **○** → 0₁₀ for you to verify.



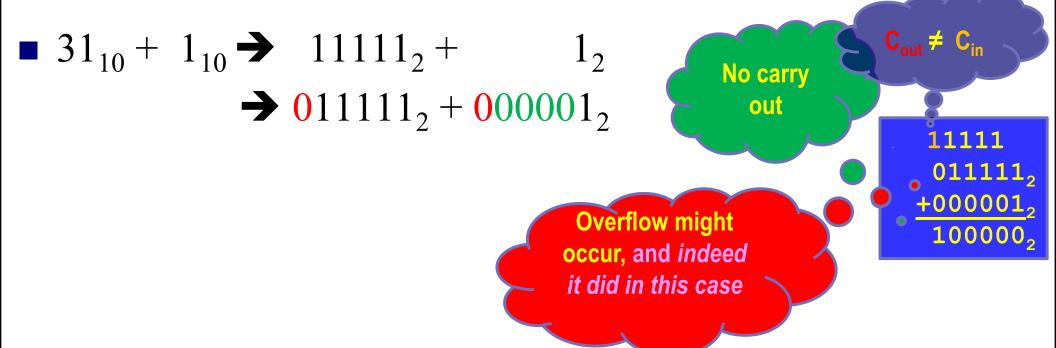
This is the answer

This step is not

■ *Example 8*:

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

- \blacksquare 31₁₀ \rightarrow 11111₂



■ *Example 9*:

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

- \blacksquare 31₁₀ \rightarrow 111111₂

Carry out to be ignored

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

$$\rightarrow$$
 (-0111111₂) + (-000001₂)

This is the answer in 2's complement

This step is not

needed. It is just,

for vou to verify.

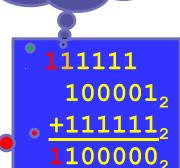
$$\rightarrow$$
 (100001₂) + (111111₂)

$$\rightarrow$$
 100000₂

$$-100000_2$$

$$\rightarrow$$
 -32₁₀

Overflow might occur, but it did not in this case



© Mahmoud R. El-Sakka

CS 2208: Introduction to Computer Organization and Architecture

■ *Example 10*:

Encode –3.25₁₀ using 2's complement 6-bit system

- \blacksquare 3.25₁₀ \rightarrow 11.01₂
- $-3.25_{10} \rightarrow -0011.01_2$
 - **→** 1100.11₂



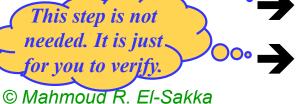
You can also look at it as if it is -3_{10} –0.25₁₀

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

This is the answer in 2's complement

- \rightarrow (111101₂) + (1111.11₂)
- **→** 111100.11₂

14



 1100.11_2 -3.25_{10}

Overflow might occur, but it did not in this case

Binary points
MUST be
aligned

111111

111101.00,

•+111111.11₂

1111100.11,

CS 2208: Introduce puter Organization and Architecture