

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 02: Signed Numbers

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

Signed Numbers

- ❑ Computer designers have adopted various techniques to represent negative numbers, including
 - *sign and magnitude*,
 - *biased representation*, and
 - *two's complement*.

Sign and Magnitude

- Example 1-a: Convert -743_8 to binary using *sign and magnitude* method

743_8

→ $111\ 100\ 011_2$

→ 111100011_2

unsigned
value

0	=	000
1	=	001
2	=	010
3	=	011
4	=	100
5	=	101
6	=	110
7	=	111

-743_8

→ 1111100011_2

Sign and Magnitude

- Example 1-b: Convert $+743_8$ to binary using *sign and magnitude* method

743_8

→ $111\ 100\ 011_2$

→ 111100011_2

unsigned
value

$+743_8$

→ 0111100011_2

0	=	000
1	=	001
2	=	010
3	=	011
4	=	100
5	=	101
6	=	110
7	=	111

Sign and Magnitude

- Example 2-a: Convert $-AB.BA_{16}$ to binary using *sign and magnitude* method

$AB.BA_{16}$

→ $1010\ 1011.1011\ 1010_2$

→ 10101011.1011101_2

unsigned
value

$-AB.BA_{16}$

→ 110101011.1011101_2

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Sign and Magnitude

- Example 2-b: Convert $+AB.BA_{16}$ to binary using *sign and magnitude* method

$AB.BA_{16}$

→ $1010\ 1011.1011\ 1010_2$

→ 10101011.1011101_2

unsigned
value

$+AB.BA_{16}$

→ 010101011.1011101_2

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Sign and Magnitude

- Example 3-a: Convert $-0.0A_{16}$ to binary using *sign and magnitude* method

$0.0A_{16}$

→ $0000.0000\ 1010_2$

→ 0.0000101_2

$-0.0A_{16}$

→ 10.0000101_2

unsigned
value

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Sign and Magnitude

- Example 3-b: Convert $+0.0A_{16}$ to binary using *sign and magnitude* method

$0.0A_{16}$

→ $0000.0000\ 1010_2$

→ 0.0000101_2

$+0.0A_{16}$

→ 00.0000101_2

unsigned
value

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Biased Representation

- Example 4: Encode -14_{10} using *excess-32* representation method (a.k.a. *biased representation*)

To encode a number using *excess-32* method, you need to add 32 to that number.

- $-14_{10} + 32_{10} = 18_{10}$

- 18_{10} is the *excess-32* representation of -14_{10}

To decode an *excess-32* value to its original value, you need to subtract 32.

- $18_{10} - 32_{10} = -14_{10}$

Biased Representation

- Example 5: Encode 14_{10} using *excess-127* representation method (a.k.a. *biased representation*)

To encode a number using *excess-127* method, you need to add 127 to that number.

- $14_{10} + 127_{10} = 141_{10}$

- 141_{10} is the *excess-127* representation of 14_{10}

To decode an *excess-127* value to its original value, you need to subtract 127.

- $141_{10} - 127_{10} = 14_{10}$

2's Complement

□ In binary arithmetic, the *two's complement* of a number is formed by

- *Subtracting the number from 2^n .*

The *two's* complement of 01100101_2 is
 $100000000_2 - 01100101_2 = 10011011_2$

In binary system,
 the sign is encoded as:
 MSD = 0 → positive
 MSD = 1 → negative

- *Flipping (inverting) all the bits of the number and adding 1.*

The *two's* complement of 01100101_2 is
 $10011010_2 + 1_2 = 10011011_2$.

Just for the sake of completeness,
 in radix R systems,
 the sign is encoded as:
 MSD < R/2 → positive
 MSD ≥ R/2 → negative.

- *Processing all the bits of the number from the least significant bit (LSB) towards the most significant bit (MSB)*

- copying all the zeros until the first 1 is reached,
- copying that 1,
- flipping (inverting) all the remaining bits.

The *two's* complement of 01100100_2 is 10011100_2 .

The *two's* complement of 01100101_2 is 10011011_2 .

2's Complement

■ Example 6:

Convert $+AB.BA_{16}$ and $-AB.BA_{16}$ to binary using *2's complement* method

$AB.BA_{16}$

$\rightarrow 1010 \ 1011.1011 \ 1010_2$

$\rightarrow 10101011.1011101_2$

unsigned value

$+AB.BA_{16}$

$\rightarrow 010101011.1011101_2$

$-AB.BA_{16}$

$\rightarrow 101010100.0100011_2$

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

2's Complement

■ Example 7:

Convert $+0.0A_{16}$ and $-0.0A_{16}$ to binary using *2's complement* method

$0.0A_{16}$

→ $0000.0000\ 1010_2$

→ 0.0000101_2

$+0.0A_{16}$

→ 00.0000101_2

$-0.0A_{16}$

→ 11.1111011_2

unsigned
value

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Signed Numbers

Binary pattern	Unsigned	Signed-and-magnitude	2's complement	Excess-8
0000	0	+0	+0	-8
0001	1	+1	+1	-7
0010	2	+2	+2	-6
0011	3	+3	+3	-5
0100	4	+4	+4	-4
0101	5	+5	+5	-3
0110	6	+6	+6	-2
0111	7	+7	+7	-1
1000	8	-0	-8	+0
1001	9	-1	-7	+1
1010	10	-2	-6	+2
1011	11	-3	-5	+3
1100	12	-4	-4	+4
1101	13	-5	-3	+5
1110	14	-6	-2	+6
1111	15	-7	-1	+7

For a given n bit binary pattern

What is the number of zeros for various values of n ?

What is the range for various values of n ?

Number of zeros

1

2

1

1

Range

$0 \rightarrow 2^n - 1$

$-(2^{n-1} - 1) \rightarrow 2^{n-1} - 1$

$-(2^{n-1}) \rightarrow 2^{n-1} - 1$

$-(2^{n-1}) \rightarrow 2^{n-1} - 1$

Unsigned

- Example 8: Convert 11011.11011_2 to decimal, assuming that it is an *unsigned* number.

$$11011_2 \rightarrow 27_{10}$$

$$0.11011_2 \rightarrow 0.84375_{10}$$

$$11011.11011_2 \rightarrow 27.84375_{10}$$

Another method:

$$\begin{aligned} 11011.11011_2 &= 1101111011_2 / 100000_2 \\ &= 891_{10} / 32_{10} \\ &= 27.84375_{10} \end{aligned}$$

Sign and Magnitude

- Example 9: Convert 11011.11011_2 to decimal, assuming that it is encoded using *sign and magnitude* method.

$$11011.11011_2 \rightarrow -1011.11011_2$$

$$1011_2 \rightarrow 11_{10}$$

$$0.11011_2 \rightarrow 0.84375_{10}$$

$$1011.11011_2 \rightarrow 11.84375_{10}$$

$$11011.11011_2 \rightarrow -11.84375_{10}$$

unsigned
value

Another method:

$$11011.11011_2 \rightarrow -1011.11011_2$$

$$1011.11011_2 = 101111011_2 / 100000_2$$

$$= 379_{10} / 32_{10} = 11.84375_{10}$$

$$11011.11011_2 \rightarrow -11.84375_{10}$$

unsigned
value

2's Complement

- Example 10: Convert 11011.11011_2 to decimal, assuming that it is encoded using *2's complement* method.

$11011.11011_2 \rightarrow \text{negative number}$

$11011.11011_2 \rightarrow -00100.00101_2$

$00100_2 \rightarrow 4_{10}$

$0.00101_2 \rightarrow 0.15625_{10}$

$00100.00101_2 \rightarrow 4.15625_{10}$

$11011.11011_2 \rightarrow -4.15625_{10}$

unsigned value

Another method:

$11011.11011_2 \rightarrow \text{negative number}$

$11011.11011_2 \rightarrow -00100.00101_2$

$00100.00101_2 = 0010000101_2 / 100000_2$
 $= 133_{10} / 32_{10} = 4.15625_{10}$

$11011.11011_2 \rightarrow -4.15625_{10}$

unsigned value

2's Complement

- The following numbers represent the same value, which is $+14_{10}$

- ☐ 01110₂
- ☐ 001110₂
- ☐ 0001110₂
- ☐ 00001110₂
- ☐ 000001110₂
- ☐ 0000001110₂
- ☐ ...

+14 using 5, 6, 7, 8, 9, and 10 bits

Basically, the sign is extended

- By Converting these numbers into the *2's complement*, you get

- ☐ 10010₂
- ☐ 110010₂
- ☐ 1110010₂
- ☐ 11110010₂
- ☐ 111110010₂
- ☐ 1111110010₂
- ☐ ...

-14 in 2's complement using 5, 6, 7, 8, 9, and 10 bits

Basically, the sign is extended

2's Complement

- Example 11: Convert 11011_2 to decimal, assuming that it is encoded using *2's complement* method.
- $11011_2 \rightarrow \text{negative number}$
- $11011_2 \rightarrow -00101_2$
- $00101_2 \rightarrow 5_{10}$
- $11011_2 \rightarrow -5_{10}$

2's Complement

- Example 12: Convert 1111011_2 to decimal, assuming that it is encoded using *2's complement* method.
- $1111011_2 \rightarrow \text{negative number}$
- $1111011_2 \rightarrow -0000101_2$
- $0000101_2 \rightarrow 5_{10}$
- $1111011_2 \rightarrow -5_{10}$

2's Complement

- Example 13: Convert 11111011_2 to decimal, assuming that it is encoded using *2's complement* method.
- $11111011_2 \rightarrow \text{negative number}$
- $11111011_2 \rightarrow -000000101_2$
- $000000101_2 \rightarrow 5_{10}$
- $11111011_2 \rightarrow -5_{10}$