

Digital Systems

Prepared by Dr. Alaa Eldeen Sayed Ahmed

IT131
Binary Number Systems

Lecture 1

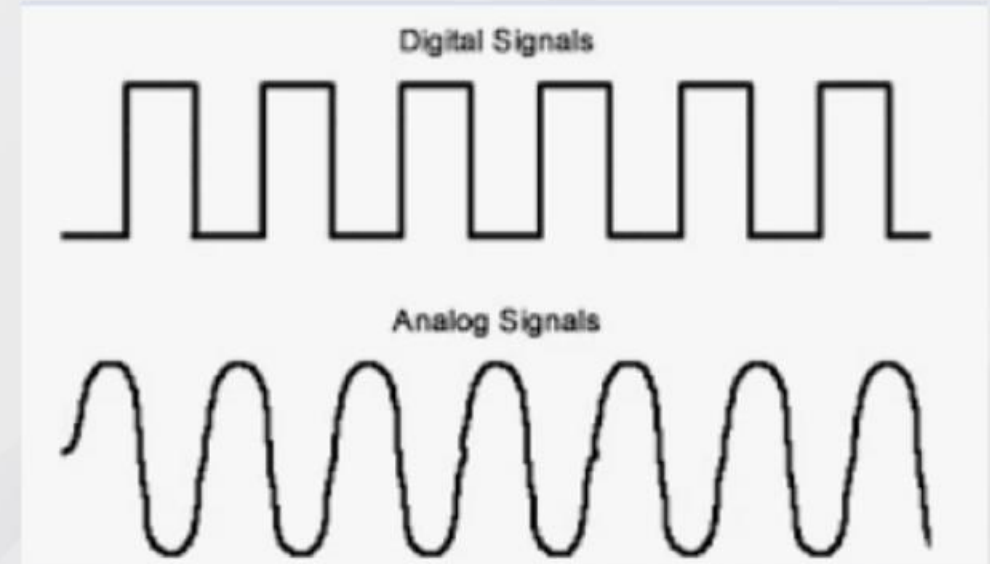


Binary Number Systems

- Objectives:
 - Digital Vs Analog
 - Numbering Systems [binary, octal and hexadecimal]
 - Base Conversion
 - Information Representation
 - Binary Additions
 - Complement

Binary Number Systems

- **Analog Signal**
 - Continuous signal that varies in amplitude and frequency.
 - Represents real-world physical quantities, like **sound** or **light**.
- **Digital Signal**
 - Discrete signal represented by binary (0s and 1s).
 - Used in digital devices like **computers** and **smartphones**.



Binary Number Systems

- **Analog Signal**
 - Infinite possible values within a range.
 - Sensitive to noise, distortion, and signal degradation.
- **Digital Signal**
 - Finite set of values (typically two: 0 and 1).
 - More resilient to noise and easier to store, transmit, and process.



Binary Number Systems

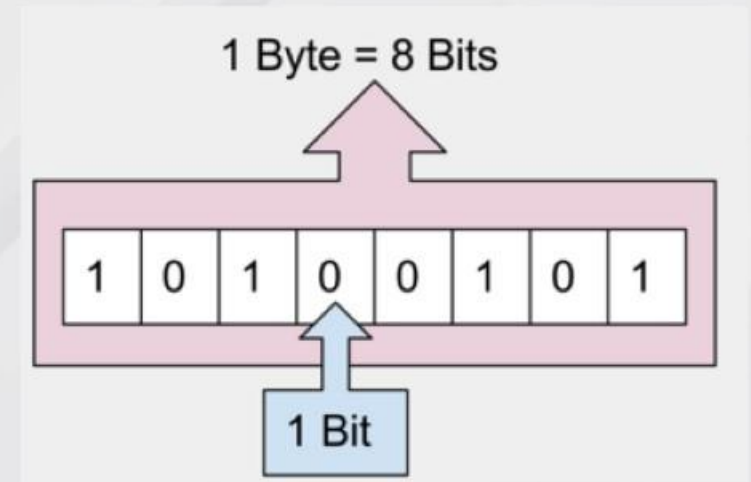
- Computers only deal with binary data (**0s and 1s**), hence all data manipulated by computers must be represented in binary format.
- Machine instructions manipulate many different forms of data:
 - Numbers:
 - Integers: 33, +128, -2827
 - Real numbers: 1.33, +9.55609, -6.76E12, +4.33E-03
 - Alphanumeric characters (letters, numbers, signs, control characters): examples: A, a, c, 1, 3, ", +, Ctrl, Shift, etc.

Binary Number Systems

- Images (still or moving): Usually represented by numbers representing the Red, Green and Blue (RGB) colors of each pixel in an image,
- Sounds: Numbers representing sound amplitudes sampled at a certain rate.
- So, in general all these data types need to be represented in binary to be handled by computer systems.

Binary Number Systems

- In binary, we have bytes and bits which are major components in writing binary.
- **Bit**: a single 0 or 1 in a binary number
 - For Example: 10100000
- **Byte**: eight bits put together
 - For Example: 10100000



Binary Number Systems

Base Conversion

Base Conversion: While computers use binary, humans often use other number systems such as decimal or hexadecimal. Knowing how to convert between these bases is important for convenience and communication.

Binary Number Systems

Converting Binary to Decimal

- The Process : Weighted Multiplication
 - a) Multiply each bit of the Binary Number by it corresponding bit-weighting factor (i.e. Bit-0 $\rightarrow 2^0=1$; Bit-1 $\rightarrow 2^1=2$; Bit-2 $\rightarrow 2^2=4$; etc).
 - b) Sum up all the products in step (a) to get the *Decimal Number*.

Example:

Convert the binary number 0111_2 into its decimal equivalent

$$(0111)_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

$$(10101001)_2 = (1 \times 2^7) + (1 \times 2^5) + (1 \times 2^3) + (1 \times 2^0) = 128 + 32 + 8 + 1 = (169)_{10}$$

Binary Number Systems

Binary Numbers

Table 1-3 Binary Bit Position Values.

2^n	Decimal Value	2^n	Decimal Value
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768

Binary Number Systems

Convert Decimal to Binary

- Repeatedly divide the decimal integer by 2. Each remainder is a binary digit in the translated value:

Division	Quotient	Remainder
37 / 2	18	1
18 / 2	9	0
9 / 2	4	1
4 / 2	2	0
2 / 2	1	0
1 / 2	0	1

← least significant bit

← most significant bit

← stop when
quotient is zero

$$(37)_{10} = (100101)_2$$

Binary Number Systems

Another Procedure for Converting from Decimal to Binary

- **Example: Converting $(76)_{10}$ to Binary**

- The highest power of 2 less or equal to 76 is 64, hence the **seventh (MSB)** bit is 1

1	.	.	.	.	.	.
---	---	---	---	---	---	---

- Subtracting 64 from 76 we get 12.

- The highest power of 2 less or equal to 12 is 8, hence the **fourth** bit position is 1

1	0	0	1	.	.	.
---	---	---	---	---	---	---

- We subtract 8 from 12 and get 4.

- The highest power of 2 less or equal to 4 is 4, hence the **third** bit position is 1

1	0	0	1	1	.	.
---	---	---	---	---	---	---

- Subtracting 4 from 4 yield a zero, hence all the left bits are set to 0 to yield the final answer

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

Binary Number Systems

- **Exercises:**

- Convert $(10110)_2 = ??$

Solution:

- $(10110)_2 = 16 + 4 + 2 = (22)_{10}$

What are the weights of the digits with value 1?

Solution,

the weights of the digits with value 1 is: 16 , 4 , 2

- What is the largest binary number that can be expressed with 5 bits?
- 5-bit binary: 11111

Binary Number Systems

- Convert the decimal number 35 to binary

Results of repeated division by 2 (quotients are followed by remainders):

$35_{10} = 17(\mathbf{1}); 8(\mathbf{1}); 4(\mathbf{0}); 2(\mathbf{0}); 1(\mathbf{0}); 0(\mathbf{1})$

Answer: $(100011)_2$

- What is the value of the third quotient ?
- The value of the third quotient = 4

Binary Number Systems

Hexadecimal Integers

- Binary values are represented in hexadecimal.

Table 1-5 Binary, Decimal, and Hexadecimal Equivalents.

Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

Binary Number Systems

Converting Binary to Hexadecimal

- Each hexadecimal digit corresponds to 4 binary bits.
- Example: Translate the binary integer **0001 0110 1010 0111 1001 0100** to hexadecimal

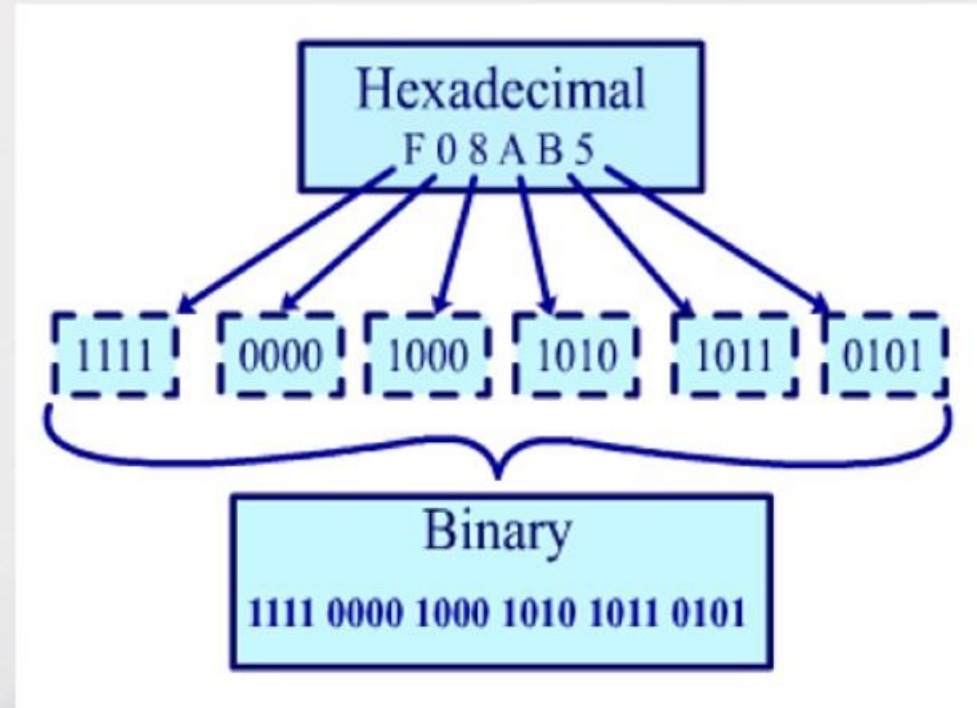
1	6	A	7	9	4
0001	0110	1010	0111	1001	0100

Binary
1 1 1 1 0 1 1
└───┘ └───┘
7 B
Hexadecimal

Binary Number Systems

Converting Hexadecimal to Binary

- Each Hexadecimal digit can be replaced by its 4-bit binary number to form the binary equivalent.



Binary Number Systems

Conversion from Binary to Octal

- Partition binary number into groups of 3 digits starting from the radix point
- When it's not a multiple of 3, you can add 0s to the left or right as you want depending on radix point
- Replace each group of 3 binary digits by the correspondent octal digit
- Example
 - $(11101010)_2 = (011\ 101\ 010)_2$
 - $= (352)_8$

Binary Number Systems

Conversion from Octal to Binary

- Replace each octal digit by the 3 corresponding binary digits. You can suppress Leading 0s
- **Example**
 - $(352)_8 = (011\ 101\ 010)_2 = (11101010)_2$
- For conversion from **octal to hexadecimal**, convert first to **binary**, then from binary to **hexadecimal**
- **Example**
 - $(352)_8 = (011\ 101\ 010)_2 = (11101010)_2$
 - $(11101010)_2 = (1110\ 1010)_2 = (EA)_{16}$
 - $(352)_8 = (EA)_{16}$

Binary Number Systems

- **Exercise**

- Convert the number $(10110)_2$ to Octal?

- $(10110)_2 = 10\ 110 = (2\ 6)_8$

- Convert the number $(1001110)_2$ to Hexadecimal?

$$(1001110)_2 = 100\ 1110 = (4\ E)_{16}$$

Binary Number Systems

- **Exercise**

- Convert the number $(372)_8$ to Binary?

- $(372)_8 = (011\ 111\ 010)_2$

- Convert the number $(10110)_2$ to Hexadecimal?

- $(372)_{16} = (0011\ 0111\ 0010)_2$

Binary Number Systems

Converting Hexadecimal to Decimal

- Multiply each digit by its corresponding power of 16:

$$\text{Decimal} = (d3 \times 16^3) + (d2 \times 16^2) + (d1 \times 16^1) + (d0 \times 16^0)$$

d = hexadecimal digit

- **Examples:**

- $(1234)_{16} = (1 \times 16^3) + (2 \times 16^2) + (3 \times 16^1) + (4 \times 16^0) = (4,660)_{10}$

- $(3BA4)_{16} = (3 \times 16^3) + (11 \times 16^2) + (10 \times 16^1) + (4 \times 16^0) = (15,268)_{10}$

Binary Number Systems

Converting Decimal to Hexadecimal

- ❖ Repeatedly divide the decimal integer by 16. Each remainder is a hex digit in the translated value:

Division	Quotient	Remainder
422 / 16	26	6
26 / 16	1	A
1 / 16	0	1

← least significant digit

← most significant digit

stop when
quotient is zero

$$(422)_{10} = (1A6)_{16}$$

Binary Number Systems

- **Exercise:**

- **Convert the number $(735)_8$ to decimal number.**

$$(735)_8 = 7 * 8^2 + 3 * 8^1 + 5 * 8^0 =$$

$$= 7 * 64 + 3 * 8 + 5 * 1 =$$

$$= 448 + 24 + 5 = 477_{10}$$

What is the value of the third term in conversion equation from right?

the value of the send term in conversion equation from right = 448

Binary Number Systems

- **Exercise:**

- **Convert the number $(FAFA)_{16}$ to decimal number.**

- $$\begin{aligned} FAFA_{16} &= 15 \cdot 16^3 + 10 \cdot 16^2 + 15 \cdot 16^1 + 10 \cdot 16^0 = \\ &= 15 \cdot 4096 + 10 \cdot 256 + 15 \cdot 16 + 10 \cdot 1 = \\ &= 61,440 + 2560 + \mathbf{240} + 160 = 64,250 \end{aligned}$$

What is the value of the second term in conversion equation from right?

the value of the send term in conversion equation from right = 240

Binary Number Systems

Convert the following numbers with the indicated bases to decimal:

a. $(4310)_5 = ??$

$$(4310)_5 = 4 * 5^3 + 3 * 5^2 + 1 * 5^1 = (580)_{10}$$

b. $(198)_{12} = ??$

$$(198)_{12} = 1 * 12^2 + 9 * 12^1 + 8 * 12^0 = 260_{10}$$

Binary Number Systems

Integer Storage Sizes

Standard sizes:

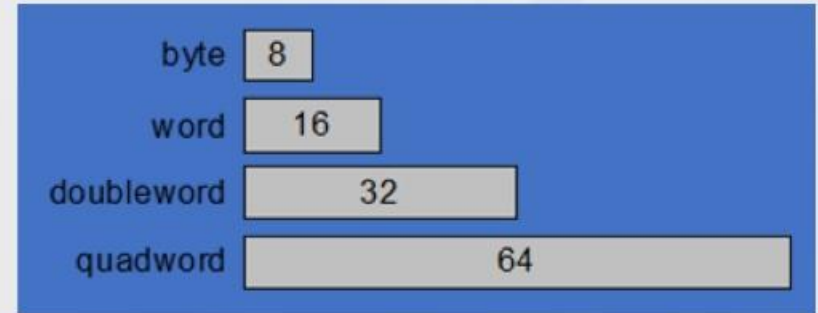


Table 1-4 Ranges of Unsigned Integers.

Storage Type	Range (low–high)	Powers of 2
Unsigned byte	0 to 255	0 to ($2^8 - 1$)
Unsigned word	0 to 65,535	0 to ($2^{16} - 1$)
Unsigned doubleword	0 to 4,294,967,295	0 to ($2^{32} - 1$)
Unsigned quadword	0 to 18,446,744,073,709,551,615	0 to ($2^{64} - 1$)

What is the largest unsigned integer that may be stored in 20 bits?

Binary Number Systems

Binary Addition

- Start with the least significant bit (rightmost bit)
- Add each pair of bits
- Include the carry in the addition, if present

carry: 1									
	0	0	0	0	0	1	0	0	(4)
+	0	0	0	0	0	1	1	1	(7)
	0	0	0	0	1	0	1	1	(11)
bit position: 7 6 5 4 3 2 1 0									

Sum Rules	
0+0 = sum 0 carry 0	
0+1 = sum 1 carry 0	
1+0 = sum 1 carry 0	
1+ 1 = sum 0 carry 1	

Binary Number Systems

Binary Addition Example

- Extending this to two multiple bit examples:

Carries	<u>0</u>	<u>0</u>
Augend	01100	10110
Addend	<u>+10001</u>	<u>+10111</u>
Sum		

- Note: The 0 is the default Carry-In to the least significant bit.

Carries	<u>00000</u>	<u>01100</u>
Augend	01100	10110
Addend	<u>+10001</u>	<u>+10111</u>
Sum	11101	101101

Binary Number Systems

- **Exercise:**
- **Add the following numbers without converting them to decimal.**

Binary numbers 1011 and 101.

```
  1011
  0101
  -----
 10000
```

Binary Number Systems

Complements in binary systems

- Complements are used for simplifying the subtraction operation and for easy manipulation of certain logical rules and events.
- Two types of complements :
 - (1's complements) and 2's complements).
- **1's complement** value obtained by inverting all the bits in the binary representation of the number (swapping 0s and 1s)
 - For example: 1's complement of 1011000 = 0100111

Binary Number Systems

- **2's complement** is obtained by starting at the LSB, leaving all the 0s unchanged, look for the first occurrence of a 1. Leave this 1 unchanged and replace each 1 by 0 and each 0 by 1 for all the bits after it.
- Another method is to add 1 to the 1's complement of the bits in the binary representation of the number.
 - For example: 2's Complement of $11010010 = (00101110)_2$
 - 1's of $11010010 = 00101101$
Then add 1 to it, $00101101 + 1 = (00101110)_2$

Binary Number Systems

- **Exercise:**

- **Obtain the 1's and 2's complements of the following binary numbers:**

00000000 → 1's complement = 11111111

2's complement = 00000000

11011010 → 1's complement = 00100101

2's complement = 00100110

10000101 → 1's complement = 01111010 → 2's complement = 01111011

Binary Number Systems

- **Exercise:**

- **Find the 2's complement of $(B2FA)_{16}$.**

- $B2FA = (1011\ 0010\ 1111\ 1010)_2 \rightarrow 2\text{'s complement} = (01001101000000110)_2$

- **Find the 2's complement of $(23)_{10}$.**

- $(23)_{10} = (10111)_2 \rightarrow 2\text{'s complement} = (01001)_2$

Binary Number Systems

Signed Integers

- Several ways to represent a signed number
 - Sign-Magnitude
 - 1's complement
 - **2's complement**
- Divide the range of values into 2 equal parts
 - First part corresponds to the positive numbers (≥ 0)
 - Second part correspond to the negative numbers (< 0)
- **Focus will be on the 2's complement representation**
 - Is Used widely in processors to represent signed integers

Binary Number Systems

Sign Magnitude Representation

- The sign-magnitude binary format is the simplest conceptual format.
- In this method of representing signed numbers, the most significant digit (MSD) **takes on extra meaning.**
- If the MSD is a 0, indicates that the number is treated as positive.
- If the MSD is a 1, this indicates that the number is negative.
- The other bits indicate the magnitude (absolute value) of the number.

Binary Number Systems

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

Binary Number Systems

One's Complement Representation of signed numbers

- In this representation:
 - **Positive numbers** are represented as standard binary numbers.
 - **Negative numbers** are represented by taking the bitwise complement (flipping all 0s to 1s and all 1s to 0s) of the corresponding positive number.

• Example 1:

• $+10 = 01010$

• $-10 = 10101$

Example 2:

$+5 = 0101$

$-5 = 1010$

Binary Number Systems

- Properties of 1's complement representation
- With the n bits, the range of numbers that can be represented in 1's complement is:
 - $[-2^{n-1}-1 \text{ to } +2^{n-1}-1]$
- **Example:**
 - If we consider 4 bits numbers, we can represent range from (-7 to +7)
 - If we consider 8 bits numbers, we can represent range from (-127 to +127)

Binary Number Systems

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

Binary Number Systems

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

Two's Complement Representation

Positive numbers

- Always Start with 0 and Signed value = Unsigned value

Negative numbers

- Always start with 1
- Start from right to left.
- Copy all bits up to and including the first 1.
- Then flip the remaining bits to the left.
- Put a negative sign in front of the binary value of the number
- Example: 10110100
- Computer the 2's comp $\rightarrow$ 01001100
- Binary value = 76
- The result = -76

Binary Number Systems

- **Exercise:**

- Represent the 1111010 using the signed-2's-complement System.

1) since the number is negative, compute the 2's complement:

0 0 0 1 1 0

2) the binary value of 0 0 0 1 1 0 is 6

3) The result is -6

Binary Number Systems

- **Exercise:**

Represent the decimal -46 and +29 to binary. using the signed-2's complement System (using 8 bits).

+ 29 $\rightarrow$ 00011101

-46 $\rightarrow$ first we get the value of 46 then compute the 2's complement

+46 = 00101110

Then -46 = 11010010

Note: the 2's complement Range= -2^{n-1} to $+(2^{n-1}-1)$

Binary Number Systems

Sign Bit

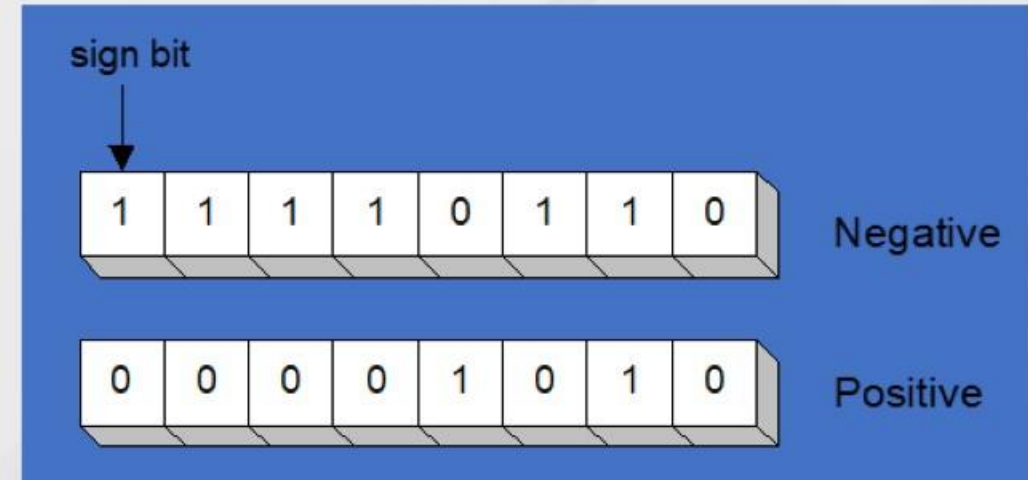
Highest bit indicates the sign. 1 = negative, 0 = positive

If highest digit of a hexadecimal is > 7 , the value is negative

Examples: 8A and C5 are negative bytes

A21F and 9D03 are negative words

B1C42A00 is a negative double-word



Binary Number Systems

Sign Extension

Step 1: Move the number into the lower-significant bits

Step 2: Fill all the remaining higher bits with the sign bit

- This will ensure that both magnitude and sign are correct

- Examples

- Sign-Extend 10110011 to 16 bits

10110011 = -77 $\rightarrow$ 11111111 10110011 = -77

- Sign-Extend 01100010 to 16 bits

01100010 = +98 $\rightarrow$ 00000000 01100010 = +98

- Infinite 0s can be added to the left of a positive number

- Infinite 1s can be added to the left of a negative number

Binary Number Systems

- **Exercise:**

Find the 2's complement of B2FA.

Convert B2FA to binary ---- 1011 0010 1111 1010

The 2's complement of B2FA = 0100 1101 0000 0110

4 D 0 6 =(4D06)₁₆

Binary Number Systems

Subtraction with 2's Complements

- The subtraction of two n-digit unsigned numbers $M-N$ in base 2 can be done as follows:
 1. Add the minuend, M , to the 2's complement of the subtrahend, N .
 2. If $M \geq N$, the end carry is discarded, what is left in the result is $M - N$.
 3. If $M < N$, there is no end carry. Then take the 2's complement of the result and place a negative sign in front.

Binary Number Systems

Example 1

Compute: $1010100 - 1000011$

1). Form 2's complement of 1000011:

1000011
0111101

2). Add the two:

1010100
0111101
(+) -----
1 0010001 (a carry)

3). Since $M \geq N$, discard the carry. Ans. = 0010001

Binary Number Systems

Example 2

Compute: $1000011 - 1010100$

1). Form 2's complement of 1010100:

1010100

0101100

2). Add the two:

1000011

0101100

(+) -----

1101111 (no carry)

3). Since $M < N$, complement the result.

Ans. = $(-)$ 0010001

Binary Number Systems

- **Exercise:**

- Perform subtraction on the unsigned binary using the 2's complement subtrahend (show the steps)

1. $10011 - 10001 \rightarrow$ first compute the 2's complement Of $10001 = 01111$

2. 10011

01111

$100010 \rightarrow$ the result = 00010

Binary Number Systems

Binary Codes

- **Binary Coded Decimal** (BCD 4-bit)
 - $(4267)_{10} = (0100\ 0010\ 0110\ 0111)_{\text{BCD}}$
 - $(9803)_{10} = (1001\ 1000\ 0000\ 0011)_{\text{BCD}}$
- The following configurations: 1010, 1011, 1100, 1101, 1110, and 1111 are not authorized in BCD
- Do NOT mix up conversion of a decimal number to a binary number with coding a decimal number with a BINARY CODE.
 - $13_{10} = 1101_2$ (conversion) $\Leftrightarrow 0001 \mid 0011$ (coding)

Binary Number Systems

- BCD Addition
- The **addition of BCD numbers** is slightly different from **binary addition**. Here, the rules of binary addition are partially applicable only to the individual 4-bit groups.
 - Get the binary sum, if $\text{sum} > 9$, add 6 to the sum
 - Obtain the correct BCD digit sum and a carry

Binary Number Systems

BCD Addition

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	1001
9	1001	12	1100	17	10001
			+0110		+0110
			10010		10111

$$184 + 576 = 760$$

BCD carry	1	1		
	0001	1000	0100	184
	+0101	0111	0110	+576
Binary sum	0111	10000	1010	
Add 6		0110	0110	
BCD sum	0111	0110	0000	760

Binary Number Systems

Weighted Codes

➤ The BCD code is the 8,4,2,1 code, and 8, 4, 2, and 1 are weights; so, BCD is a *weighted* code

Decimal	8,4,2,1	Excess3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Binary Number Systems

➤ Exercise

➤ Represent the unsigned decimal numbers 537 in BCD.

Represent the decimal number 537 in

a. BCD → 0101 0011 0111

b. Excess 3 Code. → 1000 0110 1010



Thank You