الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

كـلـيــة هندسة الحاسوب والمعلوماتية والاتصالات
Faculty of Computer & Informatics and Communications Engineering

Logic Circuits

Dr. Eng.

Hassan M. Ahmad

Hassan.Ahmad@spu.edu.sy, istamo48@mail.ru

Chapter_ 2

Number Systems, Operations and Codes

Lecture _03

Numbers, Conversions and Operations

الأعداد، التحويلات والعمليات

3-1. Decimal Numbers (الأعداد العشرية)

- The position of each digit in a weighted number system (نظام العدد الموزون/المثقل) is assigned a weight based on the **base** of the system.
  - The base of decimal numbers is **ten**, because only ten symbols (**0** through **9**) are used to represent any number.
- The column weights of decimal numbers are powers of ten that increase from right to left beginning with $10^0 = 1$:

$$\cdots 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0 .$$

- For fractional decimal (عشري كسري) numbers, the column weights are negative powers of ten that decrease from left to right:

$$\cdots\ 10^2\ 10^1\ 10^0 . 10^{-1}\ 10^{-2}\ 10^{-3} \cdots$$

(**.** -Decimal point)

- Decimal numbers can be expressed as the sum of the products of each digit times the column value for that digit.
  - Thus, the number 9240 can be expressed as

$$(9\times10^3)+(2\times10^2)+(4\times10^1)+(0\times10^0)$$

$$9000 \quad + \quad 200 \quad + \quad 40 \quad + \quad 0 \quad = 9240$$

Example 3-1

1. Express the decimal number 47 as a sum of the values of each digit.
2. Express the number 480.52 as the sum of values of each digit.

Solution

$$47 = (4\times10^1) + (7\times10^0)$$
$$= (4\times10) + (7\times1)$$
$$= 40 + 7$$

$$568.23 = (5\times10^2) + (6\times10^1) + (8\times10^0) + (2\times10^{-1}) + (3\times10^{-2})$$
$$= (5\times100) + (6\times10) + (8\times1) + (2\times0.1) + (3\times0.01)$$
$$= 500 + 60 + 8 + 0.2 + 0.03$$

3-2. Binary Numbers

- For digital systems, the binary number system is used. Binary has a **base of two** and uses the digits 0 and 1 to represent quantities.
- The column weights of binary numbers are powers of two that increase from right to left beginning with $2^0 = 1$:

$$\cdots 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0 .$$

- For fractional binary numbers, the column weights are negative powers of two that decrease from left to right:

$$\cdots\ 2^2\ 2^1\ 2^0 .\ 2^{-1}\ 2^{-2}\ 2^{-3} \cdots$$

- A binary counting sequence for numbers from **zero to fifteen (15)** is shown.
  - Notice the pattern of zeros and ones in each column.
  - Notice for four bits are required from zero to 15.

Decimal Number	Binary Number			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

In general, with ***n*** bits you can **count up to** a number equal to

$$\text{Largest decimal number} = 2^n - 1$$

For example, with five bits ($n = 5$) you can count from zero to thirty-one. With six bits (n = 6) you can count from zero to sixty-three.

$$2^5 - 1 = 32 - 1 = 31 \qquad 2^6 - 1 = 64 - 1 = 63$$

Binary weights.

Positive Powers of Two (Whole Numbers)								
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Negative Powers of Two (Fractional Number)					
2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
1/2 0.5	1/4 0.25	1/8 0.125	1/16 0.625	1/32 0.03125	1/64 0.015625

3-3. Binary Conversions (التحويلات الثنائية)

- **Binary-to-decimal Conversion**.

The decimal value of a binary number can be determined by adding the column values of all of the bits that are 1 and discarding all of the bits that are 0.

Example 3-2

Convert the binary numbers 1101101 and 0.1011 to decimal.

Solution

Determine the weight of each bit that is a 1, and then find the sum of the weights to get the decimal number.

Weight:	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary number:	1	1	0	1	1	0	1

$$1101101 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0$$
$$= 64 + 32 + 8 + 4 + 1 = \mathbf{109}$$

Weight:	2^{-1}	2^{-2}	2^{-3}	2^{-4}
Binary number:	0 . 1	0	1	1

$$0.1011 = 2^{-1} + 2^{-3} + 2^{-4}$$
$$= 0.5 + 0.125 + 0.0625 = \mathbf{0.6875}$$

❑ Decimal-to-Binary Conversion.

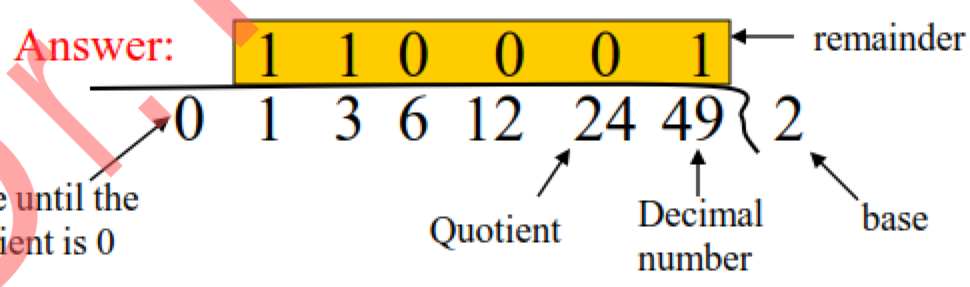
You can convert decimal to any other base by repeatedly dividing by the base. For binary, repeatedly divide by 2:

Example 3-3

Convert the decimal numbers 49 to binary by repeatedly dividing by 2.

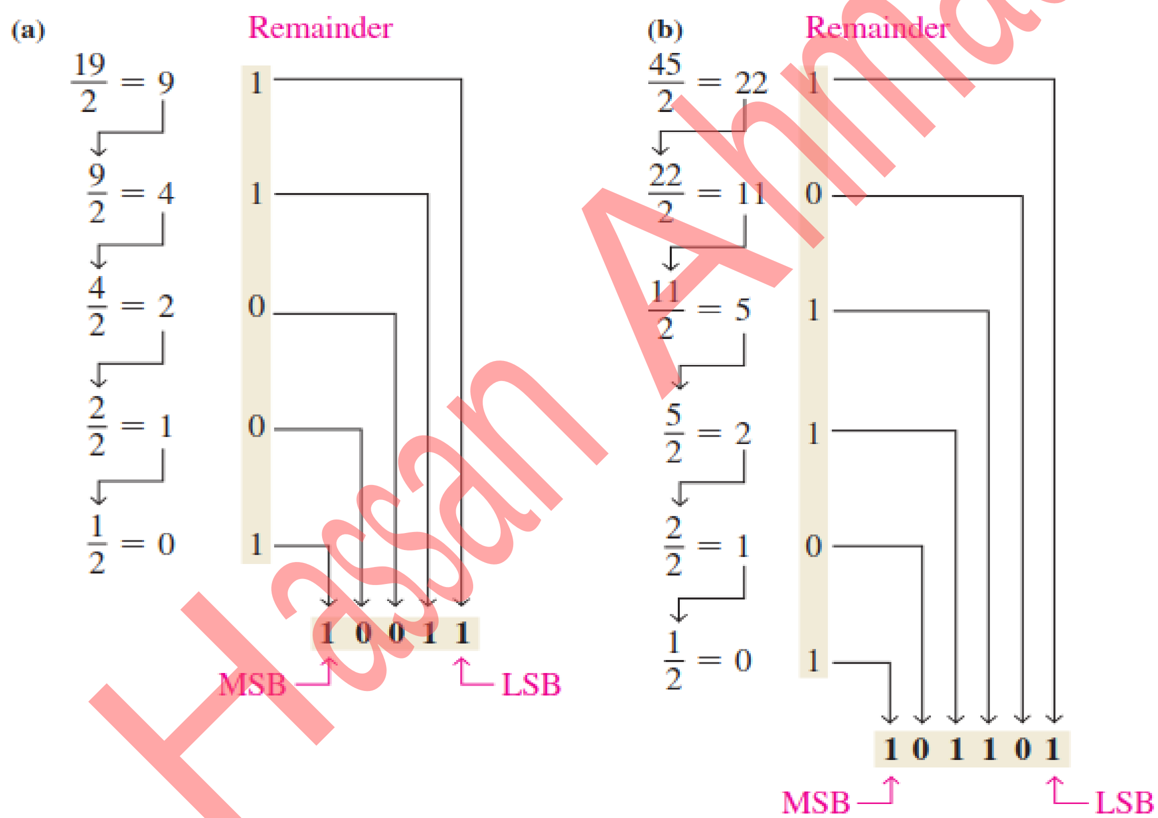
Solution

You can do this by "reverse division" and the answer will read from left to right. Put quotients (حاصل القسمة) to the left and remainders (الباقي) on top.



Example 3-4 Convert the decimal numbers 19 and 45 to binary by repeatedly dividing by 2.

Solution



LSB –Least Significant Bit (البت الأصغر منزلة، وهوالبت الموجود في أقصى يمين الرقم);

MSB – Most Significant Bit (البت الأعلى منزلة، وهوالبت الموجود في أقصى يسار الرقم).

Converting Decimal fraction to binary

You can convert a decimal fraction to binary by repeatedly multiplying the fractional results of successive multiplications by 2. The carries form the binary number.

Example 3-5 Convert the decimal fraction 0.188 and 0.3125 to binary by repeatedly multiplying the fractional results by 2.

Solution

0.188 x 2 = 0.376	carry = 0	MSB
0.376 x 2 = 0.752	carry = 0	
0.752 x 2 = 1.504	carry = 1	
0.504 x 2 = 1.008	carry = 1	
0.008 x 2 = 0.016	carry = 0	

Answer = .00110 (for five significant digits)

	Carry
$0.3125 \times 2 = 0.625$	0
$0.625 \times 2 = 1.25$	1
$0.25 \times 2 = 0.50$	0
$0.50 \times 2 = 1.00$	1

MSB → .0 1 0 1 ← LSB

Continue to the desired number of decimal places or stop when the fractional part is all zeros.

نتابع حتى العدد المطلوب من الخانات العشرية، أو نتوقف عندما يكون الجزء العشري كله 0

3-4. Binary Arithmetic Operations (العمليات الحسابية الثنائية)

❑ Binary Addition.

The **rules** for binary addition are

0 + 0 = 0	Sum = 0, carry = 0
0 + 1 = 0	Sum = 1, carry = 0
1 + 0 = 0	Sum = 1, carry = 0
1 + 1 = 10	Sum = 0, carry = 1

When an input carry = 1 due to a previous result, the rules are

Carry	Carry	
1	1	
0	1	1
+0	0	1
1	0	0

Carry bits

1 + 0 + 0 = 01	Sum of 1 with a carry of 0
1 + 1 + 0 = 10	Sum of 0 with a carry of 1
1 + 0 + 1 = 10	Sum of 0 with a carry of 1
1 + 1 + 1 = 11	Sum of 1 with a carry of 1

Example 3-6 Add the binary numbers 00111 and 10101 and show the equivalent decimal addition.

Solution

```
 0 1 1 1
 00111       7
 10101      21
 -----     ---
 11100  =   28
```

Binary Subtraction.

The **rules** for binary subtraction are

(borrow = استعارة/استقراض)

$0 - 0 = 0$

$1 - 1 = 0$

$1 - 0 = 1$

$10 - 1 = 1$ $\quad 0 - 1$ with a borrow of 1

Note: **the borrow value is $(2)_{10} = (10)_2$**

Example 3-7

Perform the following binary subtractions:

(a) 11 – 01 (b) 11 – 10

Solution

(a)

$$\begin{array}{r} 11 \\ -01 \\ \hline \mathbf{10} \end{array} \qquad \begin{array}{r} 3 \\ -1 \\ \hline 2 \end{array}$$

(b)

$$\begin{array}{r} 11 \\ -10 \\ \hline \mathbf{01} \end{array} \qquad \begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array}$$

No borrows were required in this example.

The binary number 01 is the same as 1.

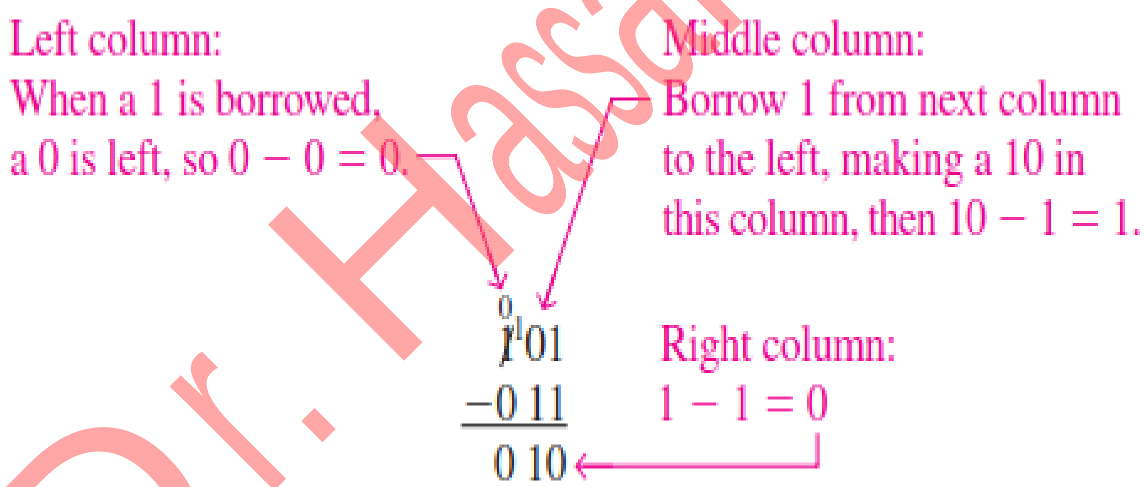
Example 3-8

Subtract the binary number 011 from 101 and show the equivalent decimal subtraction.

Solution

$$\begin{array}{r} 101 \\ -011 \\ \hline \mathbf{010} \end{array} \qquad \begin{array}{r} 5 \\ -3 \\ \hline 2 \end{array}$$

Let's examine exactly what was done to subtract the two binary numbers since a borrow is required. Begin with the right column.



Binary Multiplication.

Binary multiplication of two bits is the same as multiplication of the decimal digits 0 and 1.

The four basic rules for multiplication bits are as follows:

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Example 3-9

Perform the following multiplications:

(a) 11×11 (b) 101×111

Solution

(a)

$$\begin{array}{r} 11 \\ \times\ 11 \\ \hline 11 \\ +11 \\ \hline \mathbf{1001} \end{array} \qquad \begin{array}{r} 3 \\ \times\ 3 \\ \hline 9 \end{array}$$

Partial products: 11, +11

(b)

$$\begin{array}{r} 111 \\ \times\ 101 \\ \hline 111 \\ 000 \\ +111 \\ \hline \mathbf{100011} \end{array} \qquad \begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array}$$

Partial products: 111, 000, +111

❑ Binary Division.

Division in binary follows the same procedure as division in decimal.

Example 3-10

Perform the following divisions:

(a) 110 ÷ 11 (b) 110 ÷ 10

1	÷	1	= 1
0	÷	1	= 0
1	÷	0	Impossible
0	÷	0	Impossible

Solution

```
            10       2                    11       3
(a)   11)110      3)6       (b)   10)110      2)6
          11        6                  10        6
         000        0                   10       0
                                        10
                                        00
```

Divide $(1110111)_2$ by $(1001)_2$.

```
                          1 1 0 1         Quotient  حاصل القسمة
Divisor  1 0 0 1 | 1 1 1 0 1 1 1          Dividend  المقسوم
المقسوم عليه        1 0 0 1
                  -------
                    1 0 1 1
                    1 0 0 1
                    -------
                        1 0 1 1
                        1 0 0 1
                        -------
                            1 0   Remainder
                                  الباقي
```

3-5. Complements of Binary Numbers (متممات الأعداد الثنائية)

- The 1’s complement and the 2’s complement of binary number are important because they permit (تسمح) the representation of negative numbers.
- The method of 2’s complement arithmetic is commonly used in computers to handle negative numbers.

1’s Complement (المتمم) of Binary Numbers

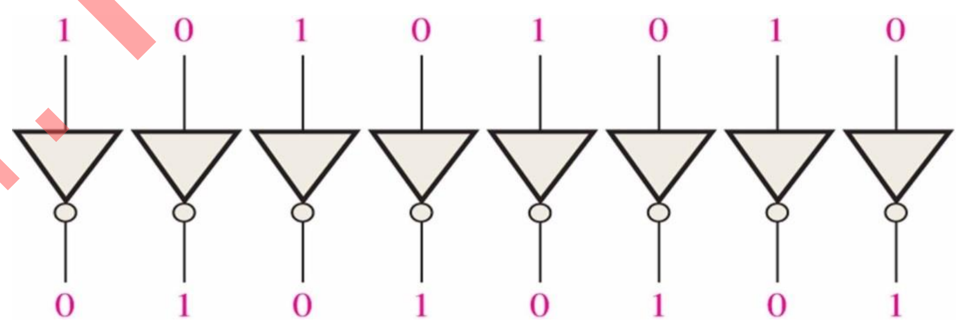
- The 1’s complement of a binary number is just the **inverse** of the digits. To form the 1’s complement, change all 0’s to 1’s and all 1’s to 0’s.
- **For example,**

1 1 0 0 1 0 1 0 Binary Number

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

0 0 1 1 0 1 0 1 1’s complement

- In digital circuits, the 1’s complement is formed by using **inverters**:



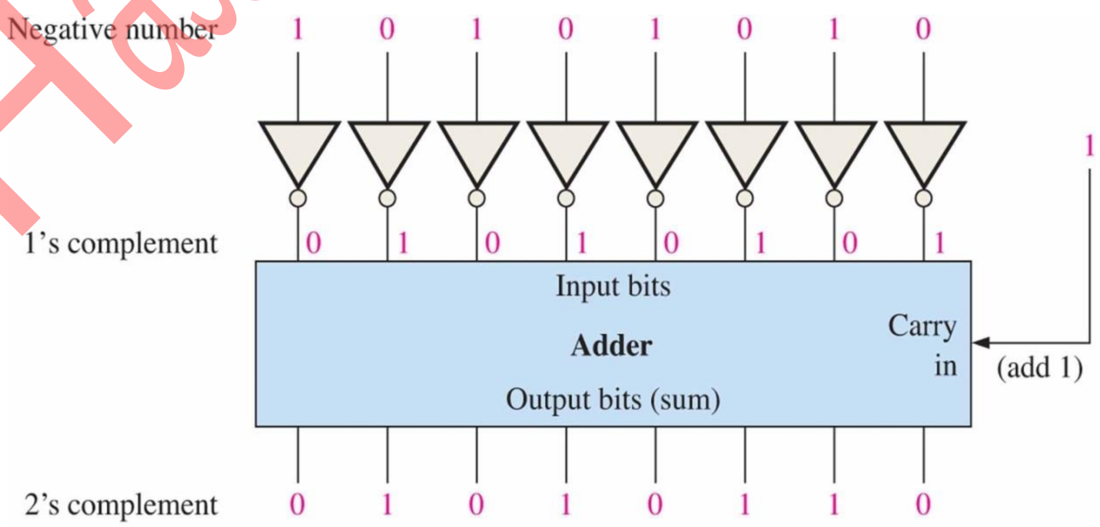
2’s Complement of Binary Numbers

- The 2’s complement of a binary number is found by **adding 1** to the LSB of the 1’s complement, i.e. $\text{2's complement} = (\text{1's complement}) + 1$

- **For example,**

```
 11001010
 00110101  (1’s complement)
       +1
 --------
 00110110  (2’s complement)
```

- In digital circuits, the process of obtaining the 2’s complement of a negative binary number is formed as



3-6. Signed Binary Numbers (الأعداد الثنائية ذات الإشارة)

- There are several ways to represent signed binary numbers. In all cases, the MSB in a signed number is the **sign bit**, that tells you if the number is **positive** or **negative**.

A 0 sign bit indicates a positive number, and a 1 sign bit indicates a negative number.

- Computers use a modified 2's complement for signed numbers. Positive numbers are stored in ***true*** form (with a 0 for the sign bit) and negative numbers are stored in ***complement*** form (with a 1 for the sign bit).

➢ **For example**, the positive number 58 is written using 8-bits as

00111010 (true form).

Sign bit — Magnitude bits

- Negative numbers are written as the 2's complement of the corresponding positive number.

➢ **For example**, the negative number – 58 is written as

–58 = 11000110 (complement form)

Sign bit — Magnitude bits

The Decimal Value of Signed binary Numbers (القيمة العشرية للأرقام الثنائية ذات الإشارة)

- Decimal values of positive and negative numbers in signed-magnitude form are determined by summing the weights in all the magnitude bit positions where there are 1s and ignoring (إهمال/تجاهل) those positions where are zeros. The sign is determined by examination of the sign bit.

Example 3-11

1) Determine the decimal value of this signed number expressed in signed magnitude: 1 0 0 1 0 1 0 1.

Solution

The seven magnitude bits and their powers-of-two weights are as follows:

		2^6	2^5	2^4	2^3	2^2	2^1	2^0	(powers-of-two)
sign	1	0	0	1	0	1	0	1	(signed number)
				16		4		1	(weights)

$16 + 4 + 1 = 21$ (sum)

The sign bit is 1, therefore, the decimal number is **– 21**.

2) Assuming that the sign bit = –128, show that 11000110 = – 58 as a 2's complement signed number.

–128	64	32	16	8	4	2	1.	
1	1	0	0	0	1	1	0	
–128	+64				+4	+2		= –58

3-7. Arithmetic Operations with Signed Numbers

- Using the signed number notation with negative numbers in 2's complement form simplifies addition and subtraction of signed numbers.

Addition

The two numbers in addition is **addend (المضاف)** and the **augend (المضاف إليه)**. The result is the **sum**.

There are **four cases** that can occur when signed binary numbers are added.

1. Both numbers positive.
2. Positive number with magnitude larger than negative number.
3. Negative number with magnitude larger than positive number.
4. Both numbers negative.

- **Rules** for **addition**:
  - Add the two signed numbers.
  - Discard (رفض) any final carries.
  - The result is in signed form.

Example 3-12

Both numbers positive.

00000111	7
+ 00000100	+ 4
00001011	11

The sum is positive and is therefor in true (uncomplemented) binary.

Both numbers negative.

	11111011	−5
	+ 11110111	+ −9
Discard carry → 1	11110010	−14

The final carry bit is discarded. The sum is negative and therefor in 2's complement form.

Positive number with magnitude larger than negative number.

	00001111	15
	+ 11111010	+ −6
Discard carry → 1	00001001	9

The final carry bit is discarded. The sum is positive and therefor in true (uncomplemented) binary.

Negative number with magnitude larger than positive number.

00010000	16
+ 11101000	+ −24
11111000	−8

The sum is negative and therefor in 2's complement form.

Subtraction

Subtraction is special case of addition. For example, subtracting +6 (subtrahend (المطروح)) from +9 (minuend (المطروح منه)) is equivalent to adding –6 to +9. Basically, *the subtraction operation changes the sign of the subtrahend and adds it to the minuend*. The result of subtraction is called the difference.

- **Rules for Subtraction**:
  - 2's complement the subtrahend and add the numbers.
  - Discard any final carries.
  - The result is in signed form.

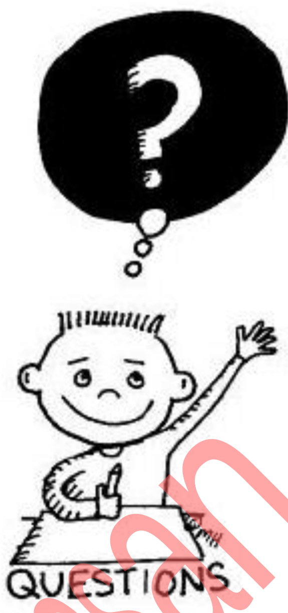
Example 3-13

$$\begin{array}{rl} & 00011110 \quad (+30) \\ - & 00001111 \quad -(+15) \\ \hline \end{array} \qquad \begin{array}{rl} & 00001110 \quad (+14) \\ - & 11101111 \quad -(-17) \\ \hline \end{array} \qquad \begin{array}{rl} & 11111111 \quad (-1) \\ - & 11111000 \quad -(-8) \\ \hline \end{array}$$

Solution

$$\begin{array}{r} 00011110 = +30 \\ 11110001 = -15 \\ \hline \not{1}\,00001111 = +15 \end{array} \qquad \begin{array}{r} 00001110 = +14 \\ 00010001 = +17 \\ \hline 00011111 = +31 \end{array} \qquad \begin{array}{r} 11111111 = -1 \\ 00001000 = +8 \\ \hline \not{1}\,00000111 = +7 \end{array}$$

Discard carry Discard carry



The end of Lecture_03, chapter 2