Lecture 8

Second Stage

Medical Physical Department

Digital Electronics

Lecture 8: Conversation System

By

Dr. Feryal Ibrahim Althefery

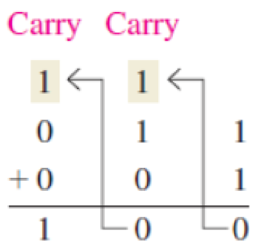
1.2 Binary Arithmetic

Binary arithmetic is essential in all digital computers and in many other types of digital systems. To understand digital systems, you must know the basics of **binary addition**, **subtraction**, **multiplication**, and **division**.

A. Binary Addition:

The four basic rules for adding binary digits (bits) are as follows:

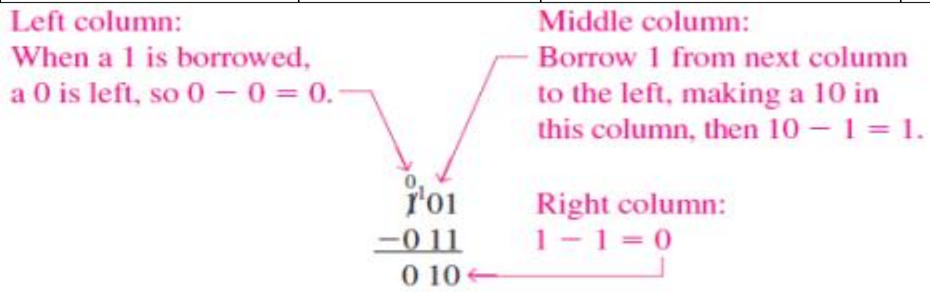
Addition		Result	Carry
0 + 0	=	0	0
0 + 1	=	1	0
1 + 0	=	1	0
1 + 1	=	0	1



B. Binary Subtraction:

The four basic rules for subtracting bits are as follows:

Input A	Input B	Subtract S = A-B	Borrow B
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0



Left column:
When a 1 is borrowed,
a 0 is left, so 0 − 0 = 0.
Middle column:
Borrow 1 from next column
to the left, making a 10 in
this column, then 10 − 1 = 1.
Right column:
1 − 1 = 0
1 0 1
−0 1 1
0 1 0

C. Binary Multiplication:

The four basic rules for multiplying bits are as follows:

Input A	Input B	Multiply (M) AxB
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} 11 \\ \times\ 11 \\ \hline \end{array} \qquad \begin{array}{r} 3 \\ \times\ 3 \\ \hline 9 \end{array} \qquad\qquad \begin{array}{r} 111 \\ \times\ 101 \\ \hline \end{array} \qquad \begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array}$$

$$\text{Partial products}\left\{\begin{array}{r} 11 \\ +11 \\ \hline 1001 \end{array}\right. \qquad\qquad \text{Partial products}\left\{\begin{array}{r} 111 \\ 000 \\ +111 \\ \hline 100011 \end{array}\right.$$

D. Binary Division:

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

Example: divide the number (11011) on (101)

```
               1 0 1
            ___________
   1 0 1  |  1 1 0 0 1
  ________|
            − 1 0 1
               -----
               0 0 1 0 1
            −      1 0 1
                   -----
                   0 0 0
```

$$\begin{array}{r} 10 \\ 11\overline{)110} \\ 11 \\ \hline 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ 6 \\ \hline 0 \end{array} \qquad\qquad \begin{array}{r} 11 \\ 10\overline{)110} \\ 10 \\ \hline 10 \\ 10 \\ \hline 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ 6 \\ \hline 0 \end{array}$$

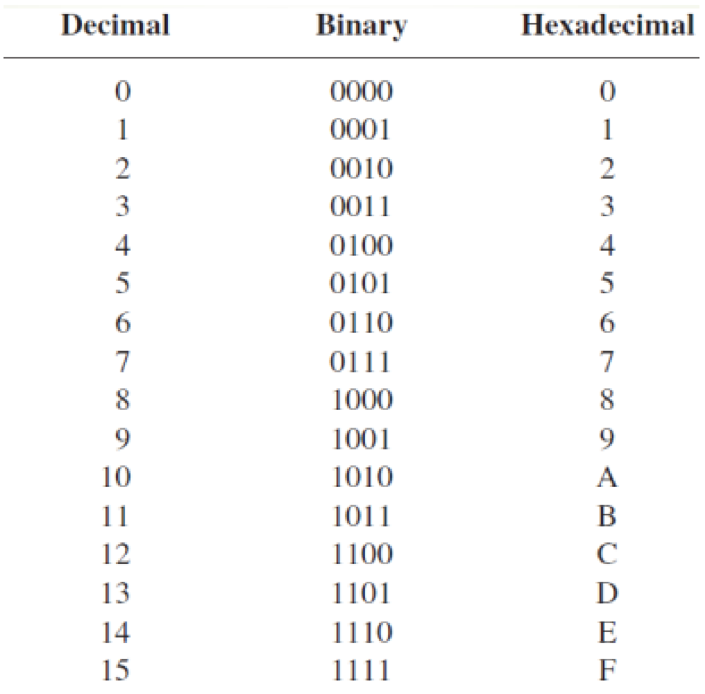
1. Hexadecimal Number

The **hexadecimal number system** is used commonly by designers to represent long strings of bits in the addresses, instructions, and data in digital systems. This system uses 16 digits to represent any quantity. The positional weight of each digit is a power of 16.

….	16^4	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	…..
….	65536	4096	256	16	1	.	$\frac{1}{16}$	$\frac{1}{256}$	$\frac{1}{4096}$	….

Each hexadecimal digit represents a 4-bit binary number (as listed in Table 1).

Table 1: Table of Number Systems



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

Example: $(7EA.8F)_{16} = 7 \times 256 + 14 \times 16 + 10 \times 1 + 8 \times \frac{1}{16} + 15 \times \frac{1}{256}$

3.1 Counting in Hexadecimal

How do you count in hexadecimal once you get to F? Simply start over with another column and continue as follows:

0, 1, 2, 3, 4, 5, 6, 7, 8, **9**, A, B, C, D, E, F,

10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
1A, 1B, 1C, 1D, 1E, 1F,

20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B,
2C, 2D, 2E, 2F,

30, 31, ….

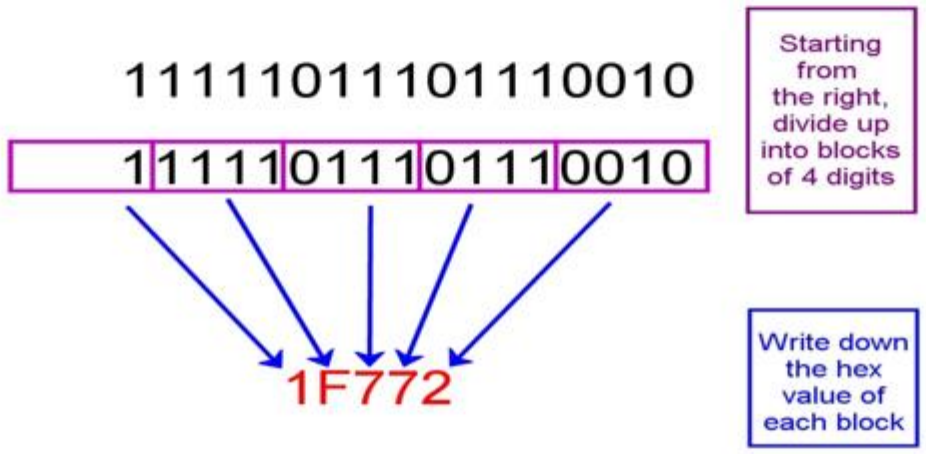
With two hexadecimal digits, you can count up to FF_{16}, which is decimal 255. To count beyond this, three hexadecimal digits are needed. For instance, 100_{16} is decimal 256, 101_{16} is decimal 257, and so forth.

The maximum 3-digit hexadecimal number is FFF_{16}, or decimal 4095. The maximum 4-digit hexadecimal number is $FFFF_{16}$, which is decimal 65,535.

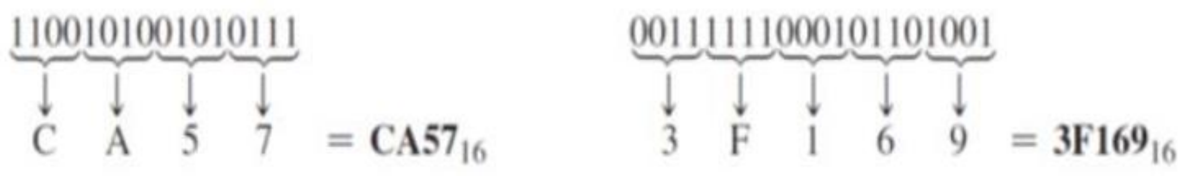
3.2 Number Base Conversion

A. Binary-to-Hexadecimal Conversion

Examples


11111011101110010
1 1111 0111 0111 0010
1F772
Starting from the right, divide up into blocks of 4 digits
Write down the hex value of each block

$$11111011101110010_2 = 1F772_{16}$$



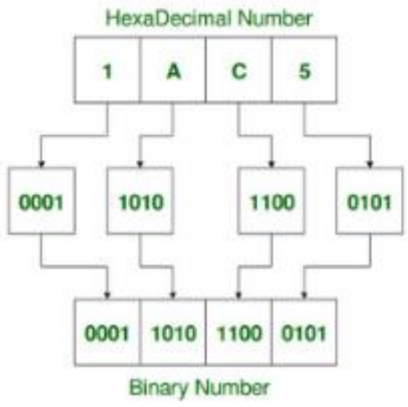
B. Hexadecimal-to-Binary Conversion

Hexadecimal-to-Binary Conversion: Replace each hexadecimal symbol with the appropriate four bits.

Examples

(a) $\overbrace{1}^{1}\overbrace{0000}^{0}\overbrace{1010}^{A}\overbrace{0100}^{4}$ (b) $\overbrace{1100}^{C}\overbrace{1111}^{F}\overbrace{1000}^{8}\overbrace{1110}^{E}$ (c) $\overbrace{1001}^{9}\overbrace{0111}^{7}\overbrace{0100}^{4}\overbrace{0010}^{2}$

(b)



C. Hexadecimal-to-Decimal Conversion

1. **Method One:** First convert the **hexadecimal number to binary** and then convert from **binary to decimal.**

(a) $\overbrace{0001}^{1}\overbrace{1100}^{C} = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28}_{10}$

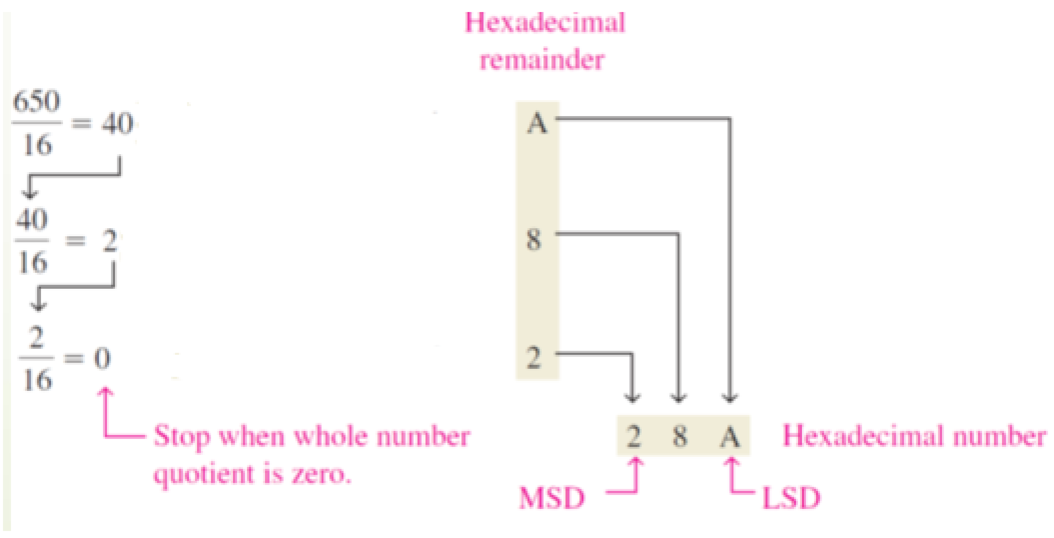
(b) $\overbrace{1010}^{A}\overbrace{1000}^{8}\overbrace{0101}^{5} = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = \mathbf{2693}_{10}$

2. **Method Two:** Multiply the decimal value of each hexadecimal digit by its weight and then take the sum of these products. For a **4-digit** hexadecimal number, the weights are:

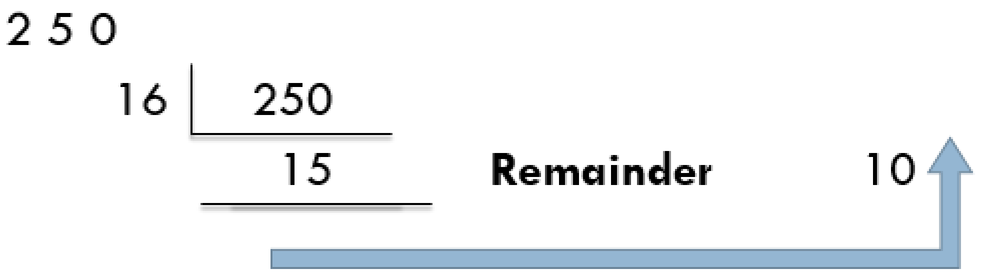
   $B2F8_{16}$ = (B × 4096) + (2 × 256) + (F × 16) + (8 × 1) = (11 × 4096) + (2 × 256) + (15 × 16) + (8 × 1) = 45,056 + 512 + 240 + 8 = 45,81610

D. Decimal-to-Hexadecimal Conversion (Repeated Division by 16)

Repeated **division by 16** will produce the equivalent hexadecimal number, formed by the remainders of the divisions. The **first remainder** produced is the least significant digit (**LSD**)



Example : 250



$$250_{10} = 15 \ 10_{16} \ ?$$
$$= FA_{16}$$

4. Octal Numbers

Like the hexadecimal number system, the octal number system provides a convenient way to express binary numbers and codes. However, it is used less frequently than hexadecimal in conjunction with computers and microprocessors to express binary quantities for input and output purposes.

The octal number system is composed of **eight digits**, which are

0, 1, 2, 3, 4, 5, 6, 7

To count above 7, begin another column and start over:

10, 11, 12, 13, 14, 15, 16, 17,

20, 21, …

4.1 Number Base Conversion

1. Octal-to-Decimal Conversion

Since the octal number system has a base of eight, each successive digit position is an increasing power of eight, beginning in the right-most column with 8^0. The evaluation of an octal number in terms of its decimal equivalent is accomplished by multiplying each digit by its weight and summing the products, as illustrated here for 2374_8

Weight: 8^3 8^2 8^1 8^0

$2374_8 = (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0)$

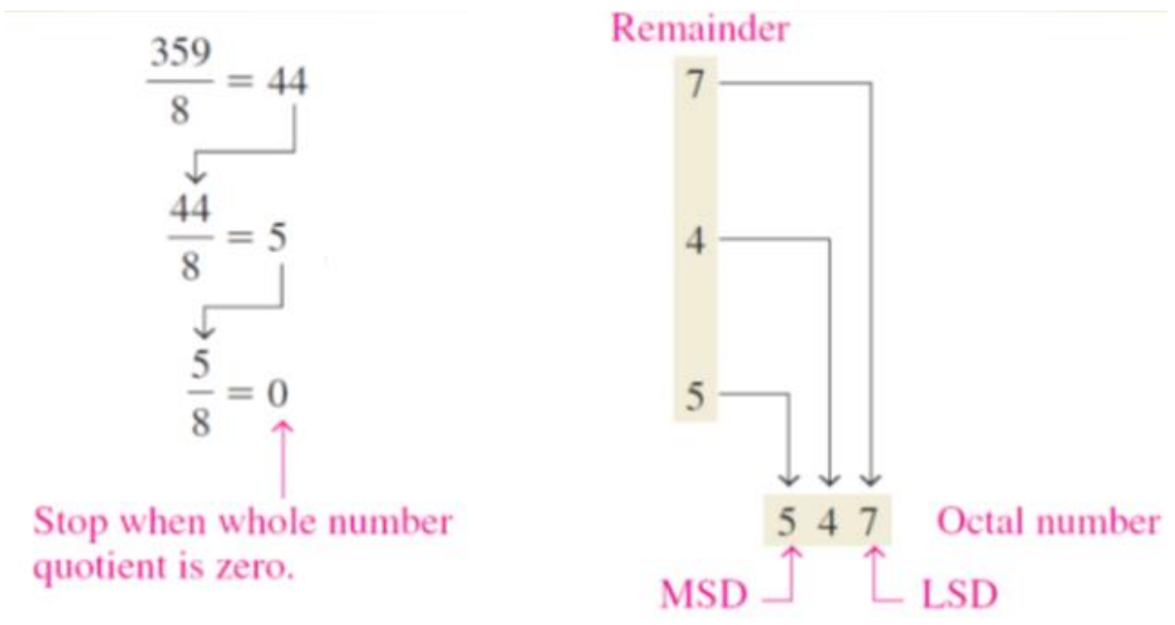
$= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1)$

$= 1024 + 192 + 56 + 4 = 127610$

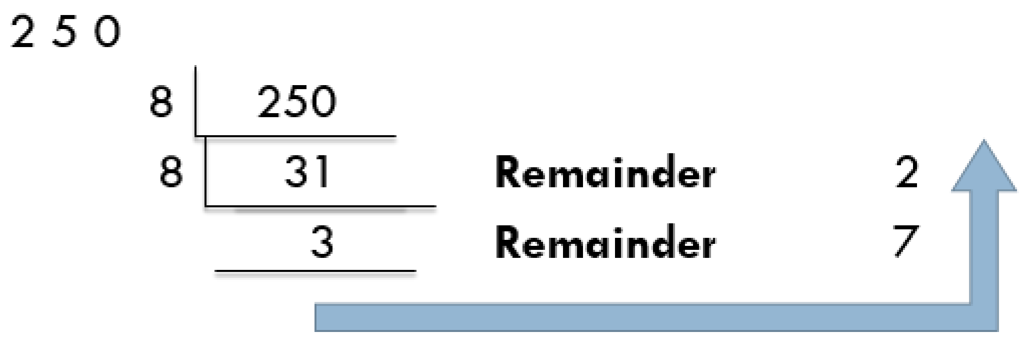
010 011 111 100

2. Decimal-to-Octal Conversion (Repeated Divisionby-8 Method)

Each successive division by 8 yields a remainder that becomes a digit in the equivalent octal number. The **first remainder** generated is the least significant digit (LSD).



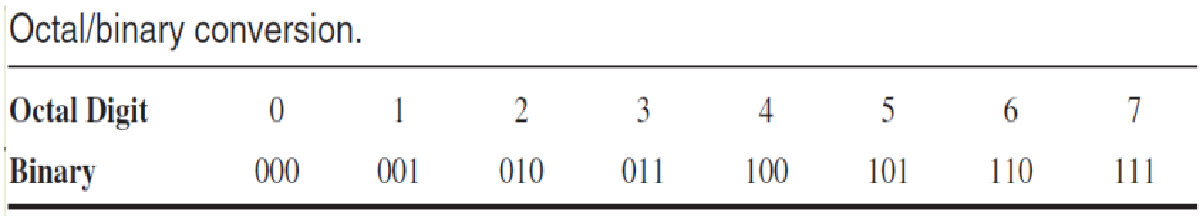
Example : 250



$250_{10} = 372_8$

3. Octal-to-Binary Conversion

Because each octal digit can be represented by a **3-bit** binary number, it is very easy to convert from octal to binary. **Each octal digit is represented by three bits** as shown in **Table below**



Octal/binary conversion.

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

(a) $\begin{array}{cc} 1 & 3 \\ \downarrow & \downarrow \\ \overbrace{001} & \overbrace{011} \end{array}$ (b) $\begin{array}{cc} 2 & 5 \\ \downarrow & \downarrow \\ \overbrace{010} & \overbrace{101} \end{array}$ (c) $\begin{array}{ccc} 1 & 4 & 0 \\ \downarrow & \downarrow & \downarrow \\ \overbrace{001} & \overbrace{100} & \overbrace{000} \end{array}$ (d) $\begin{array}{cccc} 7 & 5 & 2 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \overbrace{111} & \overbrace{101} & \overbrace{010} & \overbrace{110} \end{array}$

4. Binary-to-Octal Conversion

Start with the **right-most group of three bits** and, moving from right to left, convert each **3-bit group** to the equivalent octal digit. If there are **not three** bits available for the left-most group, **add** either one or two zeros to make a complete group. These leading zeros do not affect the value of the binary number.

(a) $\begin{array}{ccl} \underbrace{110} & \underbrace{101} & \\ \downarrow & \downarrow & \\ 6 & 5 & = \mathbf{65}_8 \end{array}$

(b) $\begin{array}{cccl} \underbrace{101} & \underbrace{111} & \underbrace{001} & \\ \downarrow & \downarrow & \downarrow & \\ 5 & 7 & 1 & = \mathbf{571}_8 \end{array}$

(c) $\begin{array}{ccccl} \underbrace{100} & \underbrace{110} & \underbrace{011} & \underbrace{010} & \\ \downarrow & \downarrow & \downarrow & \downarrow & \\ 4 & 6 & 3 & 2 & = \mathbf{4632}_8 \end{array}$

(d) $\begin{array}{ccccl} \underbrace{011} & \underbrace{010} & \underbrace{000} & \underbrace{100} & \\ \downarrow & \downarrow & \downarrow & \downarrow & \\ 3 & 2 & 0 & 4 & = \mathbf{3204}_8 \end{array}$

Table 2: Number system conversion

Decimal	Binary	Octal	Hexadecimal
0	0000	000	0000
1	0001	001	0001
2	0010	002	0002
3	0011	003	0003
4	0100	004	0004
5	0101	005	0005
6	0110	006	0006
7	0111	007	0007
8	1000	010	0008
9	1001	011	0009
10	1010	012	A
11	1011	013	B
12	1100	014	C
13	1101	015	D
14	1110	016	E
15	1111	017	F